第1章

形式语言与自动机

由于本书的大多数章节都以形式语言和自动机为主要工具，现将书中所需要用到的有关内容独立出来写成本章，作为预备知识. 对于已经了解形式语言的读者，当然无需阅读这里的介绍；对于不熟悉形式语言的读者，也建议只需对本章稍作浏览，在需要时再查阅本章的有关段节，或参考有关文献.

本章并不是关于形式语言和自动机的全面论述，内容选取完全由本书其余章节的需要决定. 所有理论性的证明也全部略去，需要了解这方面知识的读者，可参看[1].

最后，关于形式语言说几句. “形式语言”这个词看上去似乎很抽象，实际上很简单，只不过是符号串的集合. 凡是与符号串有关的研究领域，都有可能从形式语言理论中已经发展起来的概念和方法得到启发或帮助. 本书的主要内容可以看成是形式语言在动力系统研究中的应用. 当然这仅仅是开始.

§1 有限自动机与正规语言

§1.1 有限自动机的构造

可以将有限自动机设想成如图 1-1(i)所表示的一台机器，它由一条划分为许多方格的输入带和在图中用一个方框表示的控制器(或控制装置)组成. 控制器具有有限个可能状态，并在每个时刻仅处于其中的一个状态. 在这里，认为时间是离散的. 例如用 $t=0$ 代表初始时刻，用 $t=1$ 代表下一时刻，等等. 习惯上将控制器的初始状态记为 q_0. 控制器有一个读入头，即输入部件，用于

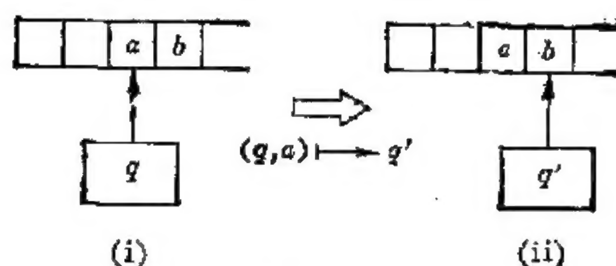


图 1-1

从输入带读入信息. 在输入带的每一个小方格中可以容纳一个符号. 这些符号均取自一个只由有限个符号组成的符号集 S. 我们称 S 为**输入符号集**. 我们还假定输入带左方有限,在 $t=0$ 时读入头的位置就在左端的第一个方格上. 以下将控制器的状态集合记为 Q. 初态 $q_0\in Q$.

控制器的功能是根据当前时刻所处的状态 q 和读入头从输入带上得到的符号 a, 确定控制器在下一时刻的状态 q', 实现这个状态的转移, 同时将输入头向右移动一个方格宽度的距离, 为扫描符号 a 右面的下一个方格作好准备. 这就是图 1-1(ii)中所示的情况. 这里不排除 $q=q'$ 的可能性. 我们也可以把"读入头向右移动"的说法改换成"输入带向左移动"的等价表述.

这里假定对每个状态 $q\in Q$ 和输入带上每个符号 $a\in S$, 控制器总是作出完全确定的反应. 我们称这样的有限自动机为**确定性有限自动机**.

控制器还具有另一个功能, 即识别功能. 假设在状态集合 Q 中有一部分状态称为**终止状态**, 它们形成 Q 的一个子集 F, $F\subseteq Q$. 控制器能够识别出当前所处的状态是终止状态, 或者不是终止状态. 我们可以将这个识别功能看成为有限自动机的输出.

现在假设在输入带上有一个有限符号串, 其符号取自符号集 S(允许重复取). 这个符号串占有从输入带左端第一个方格开始的若干个方格. 记这个符号串为 x. 控制器从状态 q_0 开始读入 x 的第一个符号, 并按上面所说的方式工作. 如果在读入 x 的最后一个符号之后, 控制器的状态转移成集合 F 中的某一个状态,

我们就称"符号串 x 为这个有限自动机所接受". 否则, 我们称" x 不为这个有限自动机所接受". 换句话说, 一个有限自动机可以识别出哪些符号串是它接受的, 哪些符号串是它不接受的.

举两个极端情况的例子. 如果在一个确定性有限自动机的状态集 Q 中, 每一个状态都是终止状态, 即 $Q=F$, 则它接受取自符号集 S 的每一个符号串. 反之, 如果在 Q 上没有任何一个状态是终止状态, 即 $F=\varnothing$(空集), 则这个有限自动机不接受任何一个符号串. 但这里的第一点对 §1.3 中更为一般的有限自动机并不成立.

我们可以将有限自动机的上述识别能力看作有限自动机的一种输出. 即对于每一个输入的符号串, 作出接受或不接受的判断, 这就是它输出的信息.

§1.2 关于形式语言的记号和概念

将一个给定的有限自动机记为 M, 我们将 M 所接受的符号串全体所成的集合称为"由 M 接受的一个形式语言", 记为 $\mathrm{L}(M)$. 对于形式语言中的符号串, 常称为语言的字(Word).

在 M 为有限自动机时, 我们称形式语言 $\mathrm{L}(M)$ 为**正规语言**(RGL). 我们已经知道两个正规语言: 一个是由取自符号集 S 的所有符号串全体所成的语言, 另一个是不含任何符号串的语言(记为 $\varnothing$, 称为空语言).

正规语言是形式语言中最简单的一类. 由于形式语言是本书的主要工具, 下面介绍形式语言方面的一些记号和概念.

在本书中, 用 S 代表由有限个不同符号组成的集合, 称为**有限符号集**. 例如{0}, {0, 1}, 等等. 其中的符号 0、1 等不作为数字来使用, 如有其他含义则另行说明.

从 S 中取有限个符号或无限个符号, 就得到符号串(在本书中, 符号串即符号序列). 如不作说明时, 我们总是考虑有限符号串, 用记号 ε 表示不含任何符号的符号串, 称为**空串**. 它在形式

语言中的作用相当于数 0 在算术中的作用. 从 S 得到的所有有限符号串的全体记为 S^*. 我们总是认为 $\varepsilon \in S^*$.

S^* 的任何一个子集,即是在符号集 S 上的一个**形式语言**. 由于 S^* 自身和空集 $\varnothing$ 也是 S^* 的子集, 因此它们都是 S 上的形式语言. 从§1.1所述, S^* 和 $\varnothing$ 都可以由有限自动机所接受, 因此它们都是正规语言. 这里要注意将 $\varnothing$ 和 ε 区别开来. $\varnothing$ 是不含任何串的一个语言,即集合论中的空集,而 ε 则是不含任何符号的空串.

设 x 和 y 是 S^* 中的两个符号串, 那么把它们连接(Catenation)得到的 xy 也是 S^* 中的一个符号串. 记号 x^i 表示将 i 个 x 连结而成的串,其中 i 是非负整数. 总认为 $x^0=\varepsilon$. 符号串 x 的长度是 x 中所含的符号个数,记为 $|x|$. 当然,有 $|\varepsilon|=0$.

如果存在串 u 和 w, 使 $x=uvw$, 那么串 v 称为串 x 的一个子串. 如果 $u=\varepsilon$,则称 v 是 x 的一个前缀(Prefix);如果 $w=\varepsilon$,则称 v 是 x 的一个后缀(Suffix). 在这三个概念中,如有 $v\neq x$,则分别称 v 为 x 的**真子串**、**真前缀**和**真后缀**.

以上概念均是对有限符号串而言的. 当然,可以将这些概念推广到无限符号串上去,这里没有什么困难. 自然,一个无限符号串的后缀仍是一个无限符号串.

§1.3 有限自动机的数学定义及其推广

一个有限自动机 M,是指一个五元组:

$$M=(Q,\ S,\ \delta,\ q_0,\ F),$$

其中 Q 是有限状态集, S 是输入符号的有限集, $q_0\in Q$ 是初始状态, $F\subseteq Q$ 是终止状态集, 这些在§1.1中都已讲过. 现在解释 δ 的含义. δ 是从 $Q\times S$ 映入到 Q 的转移函数, 也就是控制的状态转移规律. 例如,在图 1-1 中的转移可记成

$$q'=\delta(q,\ a).$$

由于 Q 和 S 都是有限集, 因此采用列表的方法就可以将函数 δ 表

示出来(表 1.1). 现在可以由 §1.1 知道, 如果 M 给定, 则正规语言 $L(M)\subseteq S^*$ 已完全确定. 反之, 如果 $L\subseteq S^*$ 是一个正规语言, 则根据定义, 一定存在一个有限自动机 M, 使 $L=L(M)$. 自然, 这样的 M 不必是唯一的.

表 1.1

	a	b	c	…
q_0				
q_1				
q_2				
⋮				

除了上面已指出 S^* 和 $\varnothing$ 是正规语言之外, 任何有限语言, 即只含有限个符号串的语言, 也都是正规语言.

前面已经提到, 上述有限自动机也称为确定性有限自动机. 现介绍几种推广.

如果控制器对于输入符号和现有状态的转移并不是完全确定的, 我们就得到非确定性有限自动机. 具体来说, 对于 $(q, a)\in Q\times S$, δ 函数的取值 $\delta(q, a)$ 可以多于一个, 也可以没有定义. 对图 1-1 而言, 对于某些 (q, a), 控制器可以转移到一个以上的状态中的某一个, 也可能对另一些 (q, a), 控制器的状态不再转移, 读入头也不再向右移动, 整个自动机处于不再工作的停机状态.

对于多值映射, 可以使用幂集合的记号. 记 2^Q 是 Q 的所有子集(包括 Q 和 $\varnothing$)所构成的集合. 那么, 非确定性有限自动机中的转移函数是从 $Q\times S$ 到 2^Q 中的映射.

另一种推广称为"带有 ε 转移的有限自动机". 这里的 ε 即是空串. 所谓 ε 转移的意义, 是允许在不读入符号时控制器的状态就可以发生转移. 用数学语言表示, δ 是定义在 $Q\times(S\cup\{\varepsilon\})$ 上的多值映射.

以上两种推广在有限自动机的理论和应用中都是很重要的手段.

我们还可以给有限自动机增加某些新的特征. 例如可以允许

读入头左移，但不移出输入带的左端，也可以允许读入头停止不动，同时具有不确定性和带 ε 转移等特征.

对于以上三种推广的有限自动机，可以相应地定义它们所接受(或识别)的形式语言，其具体细节可参看[1]. 其基本思想是自然的，即当自动机读入一个串 x 时，只要存在一个状态转移序列，使得当读完 x 后控制器能进入某一个终止状态，我们就说"x 为自动机所接受".

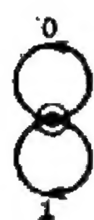
重要的是在理论上已经证明，这三种推广都没有扩大有限自动机在接受形式语言方面的能力. 这就是说，从接受语言的能力方面来看，每一种推广与确定性有限自动机都是等价的. 如果某一个语言为上述三种推广中的某一种有限自动机所接受，那么也一定可以为某一个确定性有限自动机所接受.

关于这个重要事实的证明，可参看[1]. 此外，文献[2]～[6]也都有参考价值.

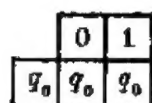
§1.4 状态转移图

有限自动机的状态转移图是很直观的一种表示方法，它是我们今后研究正规语言时的主要工具.

有限自动机的状态转移图是一个有限有向图，即由有限个结点和有向弧组成. 其中每一个结点代表自动机的一个状态，从结点到结点的有向弧(即有向线段)代表自动机的状态转移. 在有向弧上标以相应的输入符号，表示控制器由弧的起点所代表的状态在读入这个符号时转移成由弧的终点所代表的状态. 在本书中，我们用实心小圆点•代表终止状态，用空心小圆点∘代表非终止状态. 对于代表初态 q_0 的结点，则在上述圆点之外多加一个圆圈，以作区别.



(i)



(ii)

$L(M) = (0+1)^*$

(iii)

图 1-2

这样，一个有限自动机 $M=(Q, S, \delta, q_0, F)$的全部信息都在状态转移图中表示出来。

图 1-2(i)是一个最简单的确定性有限自动机。它只有一个状态，$Q=F=\{q_0\}$，$S=\{0, 1\}$。从 q_0 出发标以 0 和 1 的两条弧都回到 q_0。图 1-2(ii)是转移函数 δ 的表格表示。图 1-2(iii)则是语言 $L(M)$的正规表达式(参见§1.5)。

从代表初态 q_0 的结点出发，沿某一条有向弧到达某一个结点，然后继续这样做下去，就得到从 q_0 出发的一条路径。我们将一条路径中所含的有向弧的个数称为路径的长度。由于每一个有向弧都标有一个输入符号，因此一条路径就对应于有限自动机的一个输入符号串。例如，在图 1-2(i)中

$$q_0 \xrightarrow{0} q_0 \xrightarrow{0} q_0 \xrightarrow{1} q_0 \xrightarrow{0} q_0$$

代表长度为 4 的一条路径，它对应于输入符号串 0010。采用这样的方法，很容易在有限自动机的状态转移图上判定一个符号串 x 是否为这个自动机所接受。也就是说，只要观察从 q_0 出发与串 x 对应的路径的终点是否是一个终止状态。对于非确定性有限自动机来说，与串 x 对应的路径未必只有一条，只要存在合乎上述要求的一条路径就认为 x 为自动机所接受。图 1-2(i)的有限自动机所接受的语言显然就是 S^*，其中 $S=\{0, 1\}$。

§1.5 正规表达式

正规语言可以用正规表达式写出来，这在有时是很方便的。

先介绍构成正规表达式的三种运算。

设 L_1、L_2 和 L_3 是符号集 S 上的三个形式语言，定义 L_1 与 L_2 的和(即“并”)为

$$L_1+L_2=\{x \mid x\in L_1 \text{ 或 } x\in L_2\},$$

L_1 和 L_2 的积(即“连接”)为

$$L_1L_2=\{xy \mid x\in L_1,\ y\in L_2\}.$$

由此可以定义语言 L 的幂为

$$L^0=\{\varepsilon\}, \qquad L^i=LL^{i-1}, \qquad i\geqslant 1.$$

这里$\{\varepsilon\}$是只含 ε 的语言，它与 $\varnothing$ 不同.

第三种运算是 L 的闭包，定义为

$$L^*=\bigcup_{i=0}^{\infty} L^i.$$

现在可以用递归方式定义在 S 上的正规表达式：

1. $\varnothing$ 是正规表达式，代表空集.

2. ε 是正规表达式，代表只含空串 ε 的语言$\{\varepsilon\}$.

3. 每个符号 $a\in S$ 是正规表达式，代表只含 a 这一个串(长度为 1)的语言$\{a\}$.

4. 如果 r 和 t 是代表正规语言 R 和 T 的两个正规表达式，则$(r+t)$、(rt)和(r^*)分别是代表语言 $R+T$、RT 和 R^* 的正规表达式.

正规表达式就是由 $\varnothing$、ε 和 S 中的符号出发，有限次使用上述三种运算和括号"("、")"构成的表达式. 图 1-2(iii) 即是语言 $\{0, 1\}^*$ 的正规表达式$(0+1)^*$.

关于正规表达式的最重要结果，是它和有限自动机的等价性：每一个正规语言可以用正规表达式表示；反之，每一个正规表达式代表一个正规语言.

在文献[1]中给出了从有限自动机 M 出发求语言 L(M)的正规表达式的计算方法，也给出了从正规表达式构造相应的有限自动机的方法.

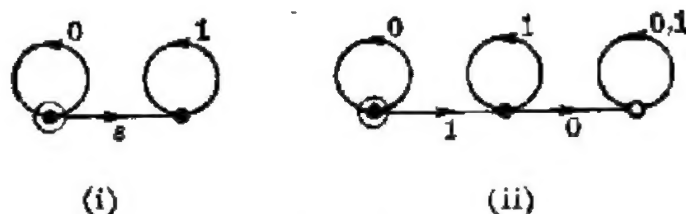


图 1-3

图 1-3 中给出了接受语言

$$L=\{0^n1^m \mid n\geqslant 0,\ m\geqslant 0\}$$

的两个有限自动机. L 的正规表达式是 0^*1^*. 其中 (i) 是带 ε 转移的非确定性有限自动机, (ii)是确定性有限自动机.

§1.6 右线性语法

正规语言还可以用某种语法规则生成. 一般的语法系统是一个四元组 $G=(V, S, P, s_0)$, 其中 V 和 S 都是有限符号集, P 是语法规则集, $s_0\in V$ 是初始符号. 设 $V\cap S=\varnothing$, 称 V 中的符号为**变量**或**非终结符**, 称 S 中的符号为**终结符**.

假定在 P 中的每一条规则都属于以下两种形式:

$$A\to\beta B \quad 或 \quad A\to\beta,$$

其中 A、$B\in V$, $\beta\in S^*$, 则我们称 G 为**右线性语法**.

从 s_0 出发, 有限次应用 P 中的规则后, 得到完全由终结符组成的串, 称为"由语法 G 生成的一个字". 所有这样的字的全体, 称为"由语法 G 生成的形式语言", 记为 $L(G)$.

这方面的基本结果是: 一个形式语言 L 为正规语言的充分必要条件是存在一个右线性语法 G, 使 $L=L(G)$.

以下举几个简单例子.

对于 $L=(0+1)^*$, 可以令 $V=\{s_0\}$, $S=\{0,1\}$, $P=\{s_0\to 0s_0, s_0\to 1s_0,\ s_0\to\varepsilon\}$.

对于 $L=0^*1^*$, 可以令 $V=\{s_0,\ s_1\}$, $S=\{0,\ 1\}$, $P=\{s_0\to 0s_0, s_0\to\varepsilon,\ s_0\to 1s_1,\ s_1\to 1s_1,\ s_1\to\varepsilon\}$.

关于左线性语法和它与正规语言的关系, 可参看[1].

§1.7 正规语言的泵引理

泵引理(Pumping Lemma)是一个形式语言为正规语言的必要条件, 它在证明某些语言不是正规语言时是很有用的. 下面叙述这个引理, 并举例说明其应用方法.

引理 如果 L 是正规语言，则存在一个只与 L 有关的整数 n，它具有下列性质：对任何 $z\in L$，且 $|z|\geqslant n$，存在 z 的一个分解 $z=uvw$，使 $|uv|\leqslant n$，$|v|\geqslant 1$，且对每个 $i\geqslant 0$ 使 $uv^iw\in L$ 成立.

［例 1］ $L=\{0^{2^i}|i\geqslant 0\}$ 不是正规语言.

用反证法. 如 L 是正规语言，则存在引理中所说的整数 n. 我们取 $z=0^{2^n}$. 因为 $|z|>n$，因此存在分解 $z=uvw$，使 $|uv|\leqslant n$，$|v|\geqslant 1$，$uv^2w\in L$. 串 uv^2w 的长度满足条件

$$2^n<|uv^2w|\leqslant 2^n+n.$$

但从 $uv^2w\in L$ 可见，长度 $|uv^2w|$ 应当是 2 的某个幂次，引出矛盾.

［例 2］ $L=\{0^i10^i|i\geqslant 1\}$ 不是正规语言.

用反证法. 如 L 为正规语言，则存在引理中所说的整数 n. 取 $z=0^n10^n$，则有分解式 $z=uvw$，使 $|uv|\leqslant n$，$|v|\geqslant 1$，$uv^iw\in L$ 对 $i\geqslant 0$ 成立. 由于 uv 完全落在 z 的前缀 0^n 中，因此当 $i\neq 1$ 时 uv^iw 不可能在符号 1 的两侧具有相同个数的符号 0. 这与 $uv^iw\in L$ 矛盾.

§1.8 自然等价关系 R_L

在正规语言研究中的另一个重要关系，是下面将介绍的自然等价关系 R_L 和迈希尔-奈罗德定理(Myhill-Nerode Theorem).

任意给定一个形式语言 $L\subseteq S^*$. 这里 L 不一定是正规语言. 利用 L，可以在 S^* 中引入一个等价关系 R_L. 设 x、$y\in S^*$，如果对每个 $z\in S^*$，串 xz 和 yz 或者都属于 L，或者都不属于 L，我们就称 x 和 y 是 R_L 等价的，记为 xR_Ly.

容易验证这样定义的 R_L 确实是一个等价关系，即满足自反性(xR_Lx)、对称性($xR_Ly\Rightarrow yR_Lx$)和传递性(xR_Ly, $yR_Lz\Rightarrow xR_Lz$). 我们称 R_L 是“由语言 L 引出的自然等价关系”.

R_L 具有右不变性质：

$$xR_Ly\Rightarrow xzR_Lyz,$$

其中 $z \in S^*$ 是任意符号串.

等价关系 R_L 将 S^* 分成若干个等价类. 记 $z \in S^*$ 所属的等价类为$[z]$. 将等价类的个数称为 R_L 的指标. 如果 R_L 的等价类有无限多个, 则称 R_L 的指标为无穷大. 指标的重要性体现在下列定理中(其证明可参看[1]):

迈希尔-奈罗德定理 设 $L \subseteq S^*$. L为正规语言的充分必要条件是 R_L 的指标为有限数. 又如 L 为正规语言, 则接受 L 的最小确定性有限自动机的状态个数就等于 R_L 的指标.

这里所讲的最小确定性有限自动机, 是指接受同一个正规语言 L 的所有确定性有限自动机中状态个数最小的一个. 今后往往简称它为最小有限自动机.

在[1]中给出了从 R_L 等价类出发构造最小有限自动机的计算方法. 同时可以证明, 接受每一个正规语言的最小有限自动机是唯一存在的. 关键的一步是用等价类作为自动机的状态, 这时R_L的定义具有明显的自动机意义. 定理的证明是容易的. 现在重新讨论上面的例子.

对于 $L=\{0^{2^i} \mid i \geq 0\}$, 任意取两个字 0^{2^i} 和 0^{2^j}, $i<j$. 令 $z=0^{2^i}$, 则 $0^{2^i}z=0^{2^{i+1}} \in L$. 但 $0^{2^j}z=0^{2^j+2^i}$, 它的指数 2^j+2^i 不是 2 的幂, 因此 $0^{2^j}z \notin L$. 根据 R_L 的定义, 可见 $0^{2^i} R_L 0^{2^j}$ 不能成立.这样我们就知道 L 中的无穷多个字分属于 R_L 的不同等价类, 因此 R_L 不是有限指标的等价关系. 从迈希尔-奈罗德定理可见 L 不是正规语言.

对于 $L=\{0^i 1 0^i \mid i \geq 1\}$, 我们考虑在$\{0, 1\}^*$中所有形状为 0^i 的串($i \geq 1$). 对于 0^i 和 0^j, $i<j$, 取 $z=10^i$ 就可看出 $0^i R_L 0^j$ 不能成立. 这样就可推出 R_L 不是有限指标的等价关系, 因而 L 不是正规语言.

与泵引理不同之处是可以用 R_L 和迈希尔-奈罗德定理来证明某些语言是正规语言, 并且可以确定接受这个语言的最小有限自动机.

§1.9 封闭性质

将正规语言类记为 $\mathscr{L}$(RG). 在某些应用中知道 $\mathscr{L}$(RG)关于哪些运算为封闭是很重要的. 以下列举这方面的几个结果，并作些说明.

1. $\mathscr{L}$(RG)关于语言的和、连接和闭包这三种运算是封闭的. 这从正规表达式的定义就可以得到证明(参见§1.5). 一般，称这三种运算为**正规运算**.

2. 将符号串 $z=a_1a_2\cdots a_n$ 反序得到的符号串 $a_n\cdots a_2a_1$ 记为 z^R，称为 z 的镜象. 对 $L\subseteq S^*$，称 $L^R=\{z^R|z\in L\}$为 L 的**镜象**. $\mathscr{L}$(RG)关于镜象运算是封闭的. 这可以从 L 的正规表达式出发来证明，只要将这个正规表达式中有关连接运算的因子顺序颠倒，就得到 L^R 的正规表达式.

3. 如 $L\subseteq S^*$，则称 $L'=S^*-L$ 是 L 的**补**. $\mathscr{L}$(RG)关于补运算是封闭的. 实际上，如有 $L=L(M)$，M 为确定性有限自动机 (Q, S, δ, q_0, F)，则只要令 $F'=Q-F$、$M'=(Q, S, \delta, q_0, F')$，就得到 $L'=L(M')$.

4. 设 $R\subseteq S^*$ 为正规语言. 对每个 $a\in S$，设 $R_a\subseteq \Delta^*$ 为正规语言. 这里 Δ 可以是与 S 不同的有限符号集. 将 R 中的每个字 $a_1a_2\cdots a_n$ 换成 $w_1w_2\cdots w_n$，其中 $w_i\in R_{a_i}$，$1\leqslant i\leqslant n$，这样就从 R 得到一个新的语言. 我们称这样的运算为**置换**. 用正规表达式容易证明 $\mathscr{L}$(RG)关于置换运算是封闭的.

5. 如果上述每个 R_a 只由一个字组成，则称这样的置换为**同态**. 从上面的结果可知 $\mathscr{L}$(RG)关于同态封闭. 同态运算在本书中将多次出现.

6. 如 h 为同态，即对每个 $a\in S$，$h(a)\in \Delta^*$. 对于语言 $L\subseteq \Delta^*$，定义

$$h^{-1}(L)=\{x|h(x)\in L\},$$

称为 L 的**逆同态象**. 可以证明 $\mathscr{L}$(RG)关于逆同态运算也是封

闭的.

7. 对于 $L\subseteq S^*$, 定义

$$\mathrm{MIN}(L)=\{x\in L \mid x \text{ 的每个真前缀不属于 } L\}.$$

利用有限自动机容易证明: $\mathscr{L}(\mathrm{RG})$关于 MIN 运算是封闭的.

§2 无限自动机

本节中将对非正规语言采用§1.1中的方法, 即用某种自动机作为接受或识别语言的装置. 随着自动机的复杂程度的增加, 可以得到越来越复杂的语言. 在下一节将介绍生成语法的方法, 它是§1.6的右线性语法的推广. 本节的内容可参看[1]~[6].

§2.1 一般性讨论

可以设想,要"设计"出接受非正规语言的自动机,无疑需要突破有限自动机的限制. 从§1知道, 有限自动机的主要特点是状态有限, 不需要记忆装置, 对于输入带上的信息只有读入的功能. 从§1.8中的自然等价关系 R_L 和迈希尔-奈罗德定理知道, 如 L 为非正规语言, 则有无限多个 R_L 等价类. 利用这个定理的证明方法(参见[1]), 用每一个等价类作为一个状态, 就可以构造出接受 L 的一个无限自动机 M. 这个自动机除了具有无限多个状态之外, 在其他方面与§1介绍的有限自动机完全一样, 而且是确定性的.

容易看出, 这样的推广在概念上虽然是简单的, 但在实际上却很困难, 因为它违反了有限描述的原则. 由于存在着无限多个可能状态, 状态转移函数 δ 一般也不能用有限描述的方法来给出.

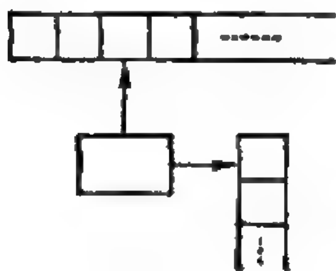
无限自动机的实际发展过程采取了与上述推广完全不同的做法. 这里所采取的方法是增强自动机的记忆能力, 增加读入头的功能等等, 但仍然保持控制器的可能状态为有限的特征. 对于每一时刻的机器动作, 也只允许做有限件事情. 这样的限制是很自然

的. 如果允许机器在有限时间内完成无限多件工作, 这并不会带来任何有实际价值的结果.

因此, 对于下面要介绍的无限自动机, 其中的所谓"无限"是一种潜在的无限. 事实上, 这种无限在有限自动机中也已存在. 例如, 在§1.1中输入带右方的长度是不受限制的. 这就是说, 虽然每一个输入符号串的长度是有限的, 但有限自动机所处理的符号串的长度是不受限制的. 从语言的识别上来看也是如此. 设 M 是一个有限自动机, $L(M)\subseteq S^*$, 则 M 能够对 S^* 中的每一个符号串作出接受或不接受的反应. 由于 S^* 中有无限多个符号串, 不可能用逐个检验的方法来求出 $L(M)$. 但如上一节所述, M 是能有限描述的. 我们可以从 M 的定义出发, 应用各种数学工具来确定它所接受的语言, 研究它的性质. 因此, 下面要介绍的各种无限自动机是在复杂程度上的增加, 而状态个数仍为有限.

§2.2 下推自动机

现在观察图2-1中的下推自动机的示意图. 与图1-1(i)中的有限自动机相比较, 主要是多了一个下推存储器和用于读写其中内容的读写头.


图 2-1

下推存储器又称堆栈(Stack), 或叫后进先出区. 它是计算机中常用的一种技术. 它的特点是, 新来的数据(即符号)总是加在原有数据上面, 而将原有内容"下推"一格; 反之, 在去掉最上面一格中的符号之后, 其余符号的位置全都上升一格. 同时, 控制器只能读出最上面一格中的符号, 或者将新的符号写入其中. 如图2-1所示, 控制器的读写头总是处在堆栈的最上面一格的位置上. 堆栈的这种工作方式也称为"后进先出", 或"先进后出". 因为较早存入到堆栈中的符号的位置总是在较晚存入的符号位置的下方, 所以, 只有当在它上面的符号均被去掉之后, 较早进入堆栈的符号才会上升到控制器的读写头可以

直接扫描的位置.

如图 2-1 所示,堆栈在下方的长度是没有限制的,从而提供了潜在无限的记忆能力. 与输入带一样,堆栈划分为方格,在每一个方格中存放一个符号. 堆栈可以使用与输入符号集不同的一个有限符号集.

与有限自动机不同, 控制器的状态转移除了与输入带上当时读入的符号和自身状态有关外, 还与当时从堆栈最上一格读入的符号有关. 此外,除了实现状态转移和读入带右移一格之外,还可以将一个符号串写入堆栈中去. 还允许在不读入输入带符号和输入头不右移的情况下对堆栈进行读写.

可以与有限自动机相仿地取定某些状态为终止态, 然后定义一个输入符号串是否为下推自动机所接受. 这样就可以进一步定义由一个下推自动机所接受(或识别)的形式语言. 我们称这样的语言为上下文无关语言(CFL), 并将上下文无关语言全体所构成的语言类记为 $\mathscr{L}(\mathrm{CF})$.

已经知道有包含关系

$$\mathscr{L}(\mathrm{RG}) \subseteq \mathscr{L}(\mathrm{CF}),$$

而且这里的包含关系为真包含. 实际上, 存在着大量的上下文无关语言,它们不是正规语言,即不能为任何有限自动机所接受. 因此可以说下推自动机的计算能力比有限自动机强, 而上下文无关语言比正规语言更为复杂.

§2.3 有两个堆栈的下推自动机

实际上有很多种方法可以进一步增强自动机的计算能力, 即识别更为复杂的语言的能力. 从下推自动机来看, 虽然它有一个堆栈作为存储器, 但堆栈的先进后出工作方式与一般的随机存取存储器相比,远为不便.

然而,只要再增加一个堆栈,如图 2-2 所示,情况就大为不同. 这里的控制器有两个读写头,分别与两个堆栈打交道. 其中,有一

个堆栈同时用作输入带，即在开始时在这个堆栈中放有输入符号串，而且将第一个符号放在堆栈的最上面的方格中，然后依次往下存放．这样，就使自动机可以按输入符号串的自然顺序读入它；但在工作中控制器还可以改写这个堆栈中的内容．因此，它不仅仅是作为输入带来使用的．

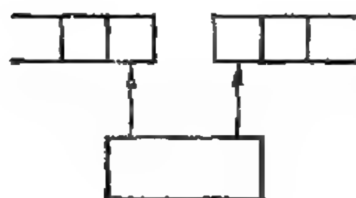


图 2-2

这里不准备进一步讨论具有两个堆栈的下推自动机．如果注意到从一个堆栈顶部方格中去掉的符号可以“下推”到第二个堆栈中去，就很容易证明具有两个堆栈的下推自动机的计算能力与下面要介绍的具有最强计算能力的图灵机(Turing Machine)相同．已经知道，增加更多的堆栈或其他存储器，都不可能得到计算能力更强的自动机．

§2.4 图灵机

在30年代中期，图灵(A. M. Turing)考虑了可计算性和自动计算机的一般定义，提出了现在被称为图灵机的概念(参见[7])．图灵机的最简单形式如图2-3所示．它与§1中介绍的有限自动机都由一个状态有限的控制器和一条划分成方格的输入带组成．但这里的控制器是通过一个读写头与输入带发生联系的．这样，它就不仅可以从方格中读入符号，而且还可以按控制器的指令将一个符号写入到这个方格中去．

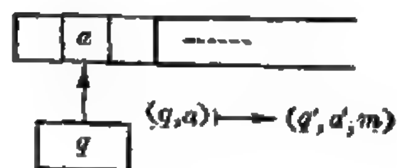


图 2-3

设输入带左方是有限的．从左端第一个方格开始依次存放输入符号串．读写头在一开始时扫描左端的第一个方格．图灵机的工作方式可以用映射 δ 来表示，它将 (q, a) 映到 (q', a', m)，其中 q 是当时的控制器状态，a 是读写头从输入带上读入的符号．映射 δ 表示图灵机根据 (q, a) 要做三件事：将控制器的状态转移成

q'，通过读写头将符号 a' 写入到符号 a 所处的方格中，决定读写头往左或往右移动一个方格．这里用 m 取 L(左)或 R(右)来代表这两种可能性．在写入过程中，如 $a=a'$，即意味着保持方格中原有的符号不变．

以上虽然是最简单形式的图灵机，但已经证明它与图灵机的其他各种形式在计算能力上是等价的．

与有限自动机相比的另一个不同之处是，图灵机不仅可以利用输入带上一开始存放有输入符号串的那一部分，而且还可以利用在输入带上的其余部分．由于输入带在右方的长度没有限制，因此就提供了无限的记忆能力．我们假定在开始时于输入符号串右方的每一个方格中都放有一个特殊的空白符号 B，并规定 B 不在输入符号集中．

与过去一样，将控制器的状态集记为 Q，并且用 F 表示在 Q 中的终止状态组成的子集，我们就可以定义由图灵机接受(或识别)的符号串和形式语言．

我们称图灵机接受的形式语言为**递归可枚举语言**(REL)，将所有递归可枚举语言构成的形式语言类记为 $\mathscr{L}$(RE)．把这样的语言叫做"可枚举"的理由，是指可以用一个图灵机将语言中所有的字逐个打印出来，即将所有字枚举出来．"递归"这个词与下面要提到的可计算性理论有关．

我们还需要知道，图灵机可以作为真正的计算机来使用．实际上，这是提出图灵机的原来目的．固定一个自然数 k，可以用图灵机来计算函数 $f(i_1, i_2, \cdots, i_k)$，其中每个自变量只取非负整数，函数值也是如此．计算的方法是在输入带上一开始放有符号串

$$0^{i_1}10^{i_2}1\cdots10^{i_k},$$

然后图灵机开始工作．如果当图灵机停机时，不论控制器是否处于终止状态，只要输入带上的符号串为 0^m，我们就定义

$$f(i_1, i_2, \cdots, i_k)=m.$$

我们称这样的函数为**部分递归函数**．显然，f 不一定对自变量的

每一组值都有定义.

这个概念的重要性可以从以下论题看出:

丘奇-图灵论题(Church-Turing Thesis) 可计算函数就是部分递归函数,即可以用图灵机计算的函数.

需要说明的是,"可计算"乃是我们的一种直观概念,并非数学定义. 因此,这个论题不可能从数学上得到证明. 反之,如果能找到一个直观上可计算的函数,并且证明它不能为任何图灵机所计算,则就可以推翻丘奇-图灵论题. 但是,至今为止,还没有发生这样的事情. 从已经尝试过的种种可计算性理论来看,都支持上述论题的正确性.

关于图灵机的另一个重要结果是,存在一类通用图灵机.如果将任何一个图灵机的有限描述 M 和输入符号串 w 用一定的方法写成一个符号串,然后输入到一个通用图灵机中去,则通用图灵机就可模拟图灵机 M 在输入 w 时的全部动作.

§2.5 递归语言与非递归可枚举语言

从部分递归函数的定义可见,当某组自变量值$(i_1, i_2, \cdots, i_k)$不属于 f 的定义域时,图灵机有可能不停机. 这样,我们就无法知道一个正在作计算的图灵机是否会最终计算出结果来. 从这个意义上说,上述可计算函数并不是能行可计算的.

如果由图灵机计算的函数 $f(i_1, i_2, \cdots, i_k)$ 对每一组自变量值都有定义,则称 f 为**全递归函数**. 这时的计算不存在上述停机困难,因此 f 是能行可计算的.

与此相应地,在递归可枚举语言类 $\mathscr{L}$(RE)中有一个真子类,称为**递归语言类**. 称一个语言 L 为**递归语言**,如果至少存在一个图灵机接受 L,同时这个图灵机对所有输入都一定会停机,其中包括接受 L 中的符号串的停机在内.

另一方面,容易说明存在非递归可枚举语言.

因为每一个图灵机的有限描述都可以按一定的方法编码成各

不相同的符号串，所以，图灵机全体只有可列个. 这表明 $\mathscr{L}(\mathrm{RE})$ 为可列集，即所有递归可枚举语言一共只有可列个.

根据"形式语言即是 S^* 的子集合"的这个定义，可知形式语言有不可列个. 这就说明，存在着大量的非递归可枚举语言，它们不可能为任何图灵机所接受. 在这个意义上，我们说这种语言的复杂性是不可计算的.

相应地，存在大量的不可计算函数，即不可能用任何图灵机来计算的函数.

§2.6 线性有界自动机

已经知道有包含关系

$$\mathscr{L}(\mathrm{CF})\subseteq\mathscr{L}(\mathrm{RE}),$$

而且是真包含关系.

实际上，图灵机的计算能力要比下推自动机强得多. 我们还需要介于这两类自动机之间的另一类无限自动机，同时介绍它所接受的一类新的形式语言.

对图灵机增加两个限制：

1° 在输入符号集中增加两个符号 ¢ 和 $，用作为输入符号串的左限制符和右限制符.

2° 控制器的读写头不允许从符号 ¢ 向左移动，也不允许从符号 $ 向右移动，也不允许改写这两个符号.

可以证明，以上限制等价于图灵机在识别一个长度为 n 的输入符号串时，所使用的存储容量不超过 n 的一个固定倍数.

我们把这样的图灵机叫做线性有界自动机，同时，把线性有界自动机所接受的形式语言叫做上下文有关语言（CSL）. 将所有上下文有关语言所构成的语言类记为 $\mathscr{L}(\mathrm{CS})$.

可以证明（参见[1]、[2]、[5]、[6]）

$$\mathscr{L}(\mathrm{RG})\subsetneq\mathscr{L}(\mathrm{CF})\subsetneq\mathscr{L}(\mathrm{CS})\subsetneq\mathscr{L}(\mathrm{RE}).$$

此外，$\mathscr{L}(\mathrm{CS})$是递归语言类的真子类.

与§1.3相仿，可以给出非确定性下推自动机、非确定性有界自动机和非确定性图灵机的定义。

关于线性有界自动机的一个未解决问题是：确定性线性有界自动机和非确定性线性有界自动机在接受（或识别）语言的能力上是否等价。在§1.3已经指出，对有限自动机来说，相应问题的回答是肯定的。已经知道，对于图灵机来说情况也是如此。但是，对于下推自动机，相应问题的回答却是否定的。我们将确定性下推自动机所接受的语言称为确定性上下文无关语言（DCFL），它构成为 $\mathscr{L}$(CF)中的一个真子类。

§3 生成语法系统

本节从生成语法系统（即短语结构语法系统）的角度介绍在§2中引进的各类非正规语言的一些基本性质。有关事实的证明，可参看[1]～[6]。

§3.1 语言的乔姆斯基层次

在§1.6中已经指出，正规语言可以用右线性语法生成。现在介绍关于非正规语言的生成语法理论。

四元组 $G=(V, S, P, s_0)$ 称为一个生成语法，或短语结构语法，其中 V 是变量（或非终结符）的有限集，S 为终结符的有限集，$V\cap S=\varnothing$，$s_0\in V$ 为初始符，P 为语法规则（即生成规则）的有限集。每一条规则的形式为 $\alpha\to\beta$，其中 α 和 β 都属于 $(V\cup S)^*$，在 α 中至少含有 V 中的一个变量。

从初始符 s_0 出发，有限次应用 P 中的语法规则所得到的符号串 x 如果完全由终结符组成，即 $x\in S^*$，就称 x 是“由语法 G 生成的一个字（或符号串）”。由 G 生成的所有字的全体，作为 S^* 的一个子集，称为语法 G 生成的语言，记为 L(G)。

在§1.6已经介绍过一种特殊的生成语法，即右线性语法。它

所生成的语言为正规语言.

现在叙述语法规则的一般应用方式.

设 $x, y \in (V \cup S)^*$. 如果存在分解式 $x = \alpha_1 \alpha \alpha_2$ 和 $y = \alpha_1 \beta \alpha_2$, 同时, 在 P 中有一条规则为 $\alpha \to \beta$, 则我们说 y 是"从 x 出发, 经过一步派生得到的", 并记为 $x \Rightarrow y$.

一般地, 对于 x、$y \in (V \cup S)^*$, 如果 $x = y$, 或者从 x 出发, 经有限步派生后得到 y, 则说 y 是从 x 派生得到的, 记为 $x \overset{*}{\Rightarrow} y$.

由此可见, 语法规则即是对符号串进行重写的规则. 规则 $\alpha \to \beta$ 即表明可将串 $x = \alpha_1 \alpha \alpha_2$ 中的子串 α 用 β 代替. 这样的派生过程一直进行到在一个符号串中没有变量出现为止. 这样得到的完全由终结符组成的符号串, 即是由该生成语法生成的字.

这样, 就可以在形式上将语言 L(G) 写成

$$\mathrm{L}(G) = \{x \mid s_0 \overset{*}{\Rightarrow} x \text{ 和 } x \in S^*\},$$

或

$$\mathrm{L}(G) = \{x \mid s_0 \overset{*}{\Rightarrow} x\} \cap S^*.$$

乔姆斯基(N. Chomsky)从研究自然语言的数学模型出发, 提出了按生成语法的复杂程度将语言划分为四个层次的理论(参见[8]、[9]).

称生成语法 $G = (V, S, P, s_0)$ 为第 i 型语法, 如果它满足以下条件:

$i = 0$: 无限制.

$i = 1$: 在每一个语法规则 $\alpha \to \beta$ 中, β 的长度不短于 α 的长度. 唯一允许的例外是可以有规则 $s_0 \to \varepsilon$, 但这时 s_0 不可以出现在任何语法规则的右方.

$i = 2$: 每一个语法规则的形式为 $A \to \beta$, 其中 $A \in V$, $\beta \in (V \cup S)^*$.

$i = 3$: 每一个语法规则的形式为 $A \to \beta B$ 或者 $A \to \beta$, 其中 A、$B \in V$, $\beta \in S^*$.

显然，第 3 型语法即是 § 1.6 中的右线性语法.

今后，我们称由第 i 型语法生成的形式语言为第 i 型语言，并将第 i 型语言全体所成的语言类记为 $\mathscr{L}_i$，i 从 0 到 3.

以上方法划分的语言层次称为乔姆斯基层次(Chomsky Hierarchy)。它在形式语言理论中起着主导作用(参见[1]～[6]).

已经知道，以上四个层次的语言类和上两节用各种自动机定义的语言类存在如下关系:

$\mathscr{L}_0=\mathscr{L}(\mathrm{RE})$，即递归可枚举语言类.

$\mathscr{L}_1=\mathscr{L}(\mathrm{CS})$，即上下文有关语言类.

$\mathscr{L}_2=\mathscr{L}(\mathrm{CF})$，即上下文无关语言类.

$\mathscr{L}_3=\mathscr{L}(\mathrm{RG})$，即正规语言类.

因此也有包含关系

$$\mathscr{L}_3\subsetneqq\mathscr{L}_2\subsetneqq\mathscr{L}_1\subsetneqq\mathscr{L}_0.$$

从 § 2.5 知道，存在不是第 0 型的语言. 实际上，每个 $\mathscr{L}_i$ (i 从 0 到 3)是可列集，而形式语言全体为不可列集.

此外，在 $\mathscr{L}_0$ 中有一个由递归语言组成的真子类，它包含 $\mathscr{L}_1$ 作为自己的真子类. 目前还不清楚递归语言的语法特征是什么. 在 $\mathscr{L}_2$ 中则有一个由确定性上下文无关语言组成的真子类，它包含 $\mathscr{L}_3$ 作为自己的真子类.

§ 3.2 上下文无关语言的例子

现在考察 $\mathscr{L}_2(=\mathscr{L}(\mathrm{CF}))$ 中的几个简单例子.

[例 1] $\mathrm{L}=\{0^i10^i \mid i\geqslant 1\}$. 在 § 1.7 中用泵引理已经证明它不是正规语言. 现在我们来证明这个 L 是上下文无关语言. 为此，只要构造出一个第 2 型语法 G，使 $\mathrm{L}=\mathrm{L}(G)$.

令 $G=(V, S, P, s_0)$，其中 $V=\{s_0\}$、$S=\{0, 1\}$、$P=\{s_0\to 010, s_0\to 0s_00\}$. 用数学归纳法容易证明 $\mathrm{L}=\mathrm{L}(G)$.

以后也称第 2 型语法为“上下文无关语法”.

为方便起见，在列举语法规则时可将 $\alpha\to\beta_1$ 和 $\alpha\to\beta_2$ 合写

为 $\alpha \to \beta_1 | \beta_2$. 对于多于两个规则的合写记法是类似的.

[例 2] 在算术表达式中,括号的使用、配对和取消可以用一个上下文无关语言来描述, 即所谓狄克语言(Dyck Language).

取终结符集

$$S = \{e_1, \cdots, e_n, e'_1, \cdots, e'_n\},$$

变量集 $V = \{s_0\}$, 语法规则集

$$P = \{s_0 \to \varepsilon | s_0 s_0 | e_1 s_0 e'_1 | \cdots | e_n s_0 e'_n\},$$

就得到 $G = (V, S, P, s_0)$ 和狄克语言 $L = L(G) \subseteq S^*$.

L 中字的特征是有限次应用 $e_i e'_i \to \varepsilon$ 之后会变成 ε(其中 $i = 1, 2, \cdots, n$). 这里不允许将 $e'_i e_i$ 变为 ε. 可以看出, 符号 e_i 和 e'_i 是对于算术表达式中左括号和右括号的模拟, 取 i 从 1 到 n 则代表有 n 种不同类型的括号混合使用.

例如, 在 $n=1$ 时, 将 e_1 和 e'_1 直接用"("和")"表示, 则下列符号串即是这时的狄克语言中的一个字:

$$((\ )((\ )(\ )))(((\ ))((\ )(\ ))).$$

§3 3 上下文无关语言的泵引理

与 §1.7 相仿, 上下文无关语言也有相应的泵引理(Pumping Lemma), 所起的作用也是类似的.

引理 如果 L 是一个上下文无关语言, 则存在一个只与 L 有关的整数 n, 它具有下列性质: 对任何 $z \in L$, 且 $|z| \geq n$, 存在 z 的一个分解 $z = uvwxy$, 使 $|vx| \geq 1$, $|vwx| \leq n$, 且对每个 $i \geq 0$ 使 $uv^i wx^i y \in L$ 成立.

其证明可参看文献[1]、[2]、[5]、[6]中的任何一种. 这里只限于举例说明它的用法.

[例 3] $L = \{0^{2^i} | i \geq 0\}$.

在 §1.7 已经证明 L 不是正规语言. 现在我们来证明 L 也不是上下文无关语言.

用反证法. 设 L 是上下文无关语言, 则按引理, 存在满足引

理中所述条件的整数 n. 任取一个 $z\in L$, $|z|\geqslant n$, 则有分解式 $z=uvwxy$, 使 $|vx|\geqslant 1$、$|vwx|\leqslant n$, 且对每个 $i\geqslant 0$ 有 $uv^iwx^iy\in L$. 这表明长度为 $|z|+(i-1)|vx|$ 的所有 0 串(即完全由符号 0 组成的串)都是 L 中的字. 但这与 L 中的所有字的长度为等比数列 $\{2^i\}_{i\geqslant 0}$ 相矛盾.

[例 4] $L=\{a^ib^ic^i \mid i\geqslant 1\}$不是上下文无关语言.

仍用反证法. 设 L 是上下文无关语言, 则存在引理中所说的整数 n. 取 $z=a^nb^nc^n$, 则有分解式 $z=uvwxy$, $|vx|\geqslant 1$、$|vwx|\leqslant n$, 且对每个 $i\geqslant 0$ 有 $uv^iwx^iy\in L$. 从 $|vwx|\leqslant n$ 可见 vwx 不可能同时含有三种符号 a、b、c. 因此, 在 $uwy(i=0)$ 中三种符号个数不相等, 引出矛盾.

§3.4 奥登引理 (Ogden Lemma)

泵引理是一个语言为上下文无关语言的必要条件, 并不是充分条件. 本书中还需要更为有力的奥登引理. 它虽然也是以必要条件的形式出现的, 但不易找到一个例子证明它不是充分条件. 其证明可看原文[11].

奥登引理 如果 L 是上下文无关语言, 则存在一个只与 L 有关的整数 n, 它具有下列性质: 对任何 $z\in L$, $|z|\geqslant n$, 并且用任意方式将符号串 z 中不少于 n 个符号的位置选为特定的可区分位置, 存在 z 的一个分解 $z=uvwxy$, 满足以下条件:

1° 或者 u 和 v 都含有可区分位置, 或者 x 和 y 都含有可区分位置.

2° vwx 至多含 n 个可区分位置.

3° w 至少含一个可区分位置.

4° 对每个 $i\geqslant 0$ 成立 $uv^iwx^iy\in L$.

从这个引理可以直接看出, 如果将符号串 z 的每个符号的位置都选为可区分位置, 则就得到 § 3.3 中的泵引理.

现在举一个例子.

[例 5]　$L=\{a^ib^jc^kd^l \mid i=0 \text{ 或 } j=k=l\}$.

用泵引理不能证明 L 不是上下文无关语言. 实际上, 对于 $z=a^ib^jc^kd^l\in L$, 如 $i=0$, 则 j、k、l 无限制, 如 $i>0$, 则在分解式 $z=uvwxy$ 中的 vwx 可放在前缀 a^i 中, 因此不能引出矛盾.

现在用奥登引理. 设 L 是上下文无关语言, 则存在引理中所说的整数 n. 取 $k>0$ 和

$$z=a^kb^{n+1}c^{n+1}d^{n+1}\in L,$$

并且将子串 c^{n+1} 所在的 $n+1$ 个位置选为可区分位置. 这时, 如果有分解式 $z=uvwxy$, 则为了满足条件 2°, vwx 就不能覆盖 c^{n+1}, 从而不可能同时含有符号 b 和 d. 因此, 当 $i>0$ 时, 在 uv^iwx^iy 中符号 b 和 d 的个数不会相等. 这与 $k>0$ 和 $uv^iwx^iy\in L$ 相矛盾.

§3.5　关于 $\mathscr{L}_2$ 和 $\mathscr{L}_3$ 的两个定理

在什么情况下一个上下文无关语言也是正规语言? 这方面, 我们介绍两个结果.

第一个结果给出了上下文无关语言也是正规语言的一个充分必要条件.

设 $G=(V, S, P, s_0)$ 为上下文无关语法. 如果在派生过程中不出现 $A\overset{*}{\Rightarrow}\alpha A\beta$, 其中 A 为变量, α 和 β 是 $(V\cup S)^*$ 中的非空串, 则称 G 具有非自嵌套性质.

观察 §3.2 中的例 1 和例 2, 从它们的语法规则已可看出, 其中的两个语法都不具有非自嵌套性质.

实际上, 这并非偶然. 关于这方面, 有下列结果:

定理 1　一个上下文无关语言 L 同时也是正规语言的充分必要条件是, 存在一个具有非自嵌套性质的上下文无关语法 G, 使 $L=L(G)$.

特别是, 在 §1.6 的右线性语法就具有非自嵌套性质.

关于定理 1 的证明, 可以看[2].

第二个结果是符号集 S 只含一个符号时的有关事实：对于只由一个符号生成的形式语言，如果是上下文无关语言，则一定也是正规语言．为了较完整地叙述这个结果，先介绍以下概念．

整数集 $\{c+pi \mid i \geqslant 0\}$ 叫做**线性集**，其中 c、p、i 均为整数．如 $p=0$，即为 $\{c\}$．有限个线性集的并称为**半线性集**．

定理 2 如果 L 是只由一个符号生成的形式语言．则以下三个条件等价：

1° L 为正规语言．

2° L 为上下文无关语言．

3° L 中字的长度所成的整数集是一个半线性集．

关于定理 2 的证明，可参看[1]、[10]．

§3.6 上下文有关语言

§3.1 给出了上下文有关语言的生成语法定义．但是，它还有另一个等价的生成语法定义．

这里，在生成语法 $G=(V, S, P, s_0)$ 的生成规则集 P 中，我们要求每个规则的形式为

$$\alpha_1 A \alpha_2 \to \alpha_1 \beta \alpha_2,$$

其中 α_1、α_2、$\beta \in (V \cup S)^*$，$A \in V$，$\beta \neq \varepsilon$．这里唯一允许的可能例外是 $s_0 \to \varepsilon$．同时，如果 P 中有这个规则，则 s_0 不能出现在任何规则的右方．

可以证明，按照这个定义得到的语言类即是 $\mathscr{L}_1(=\mathscr{L}(\mathrm{CS}))$．我们称 $\mathscr{L}_1$ 中的语言为上下文有关语言的原因，即是从这第二个定义来的．实际上，在第二个定义中，变量 A 只有在 $\alpha_1 A \alpha_2$ 的情况下才可以将 A 改写为 β 而得到 $\alpha_1 \beta \alpha_2$，也就是说，$A \to \beta$ 仅在其"上下文"为 α_1 和 α_2 时才能使用．

对于一个上下文有关语言，要写出相应的生成语法往往不是一件容易的事．下面举两个例子，其中的语法仍使用 §3.1 中关于第 1 型语法的定义．当然，我们也称第 1 型语法为上下文有关

语法.

[例 6] $L=\{0^{2^i}\mid i\geqslant 0\}$是上下文有关语言.

例 3 已经证明这个 L 不是上下文无关语言. 现在我们写出一个上下文有关语法 G, 并证明 $L=L(G)$.

令语法

$$G=(\{s_0, C, D, E, F, L, R\}, \{0\}, P, s_0),$$

其中 P 给定如下:

$$P=\{s_0\to 0\mid 0^2\mid 0^4\mid DLCE,\ DL\to DCR\mid 0^2F,$$
$$RC\to CCR,\ RE\to LCCE,\ CL\to LC,$$
$$FC\to 0^2F,\ FE\to 0^4\}.$$

由于其中每条规则都满足§3.1中关于第1型语法的要求, 因此 $L(G)$是上下文有关语言. 下面只要证明 $L=L(G)$.

在 L 中的字 0、0^2 和 0^4 可由 s_0 直接生成. 一般地则有

$$s_0 \overset{*}{\Rightarrow} DLC^{a_n}E,$$

其中 $a_0=1$. 我们要研究数列$\{a_n\}$的规律性. 从 $DLC^{a_n}E$ 出发, 分两种情况讨论.

如果不使用 $DL\to 0^2F$, 则有

$$DLC^{a_n}E\Rightarrow DCRC^{a_n}E\overset{*}{\Rightarrow} DC^{1+2a_n}RE$$
$$\Rightarrow DC^{1+2a_n}LC^2E\overset{*}{\Rightarrow} DLC^{2a_n+3}E.$$

可见 $a_{n+1}=2a_n+3$. 从初值 a_0 可求出 a_n 的通式为

$$a_n=2^{n+2}-3.$$

从 $DLC^{a_n}E$ 出发的另一种派生是使用规则 $DL\to 0^2F$. 这时有

$$s_0\overset{*}{\Rightarrow} DLC^{a_n}E\Rightarrow 0^2FC^{a_n}E$$
$$\overset{*}{\Rightarrow} 0^{2a_n+2}FE\Rightarrow 0^{2a_n+6}=0^{2^{n+3}}.$$

这对 $n=0, 1, \cdots$ 成立. 这样, 我们已证明了 $L\subseteq L(G)$. 可以证明在 $L(G)$中也只有 0^{2^i} 这种形式的字($i\geqslant 0$), 因此 $L=L(G)$(参

见[1]、[6])。

[例 7] $L=\{a^ib^ic^i|i\geqslant 1\}$.

从§3.3中的例4已经知道L不是上下文无关语言. 现在令 $G=(V, S, P, s_0)$, 其中 $V=\{s_0, X, Y\}$, $S=\{a, b, c\}$, 语法规则集为

$$P=\{s_0\to abc|aXbc,\ Xb\to bX,\ Xc\to Ybcc,$$
$$bY\to Yb,\ aY\to aaX|aa\}.$$

按照§3.1中关于 $\mathscr{L}_1(=\mathscr{L}(CS))$ 的定义, 可见 $L(G)$ 是上下文有关语言. 现在证明 $L=L(G)$.

从 s_0 出发, 除了直接生成 $abc\in L$ 之外, 对每个 $i\geqslant 1$ 可以看出有

$$s_0\overset{*}{\Rightarrow}a^iXb^ic^i.$$

为了证明这一点, 可以对 i 用数学归纳法. $i=1$ 是生成规则集 P 中的一条规则. 现设 i 已成立, 则有

$$s_0\overset{*}{\Rightarrow}a^iXb^ic^i \quad \text{(归纳假设)}$$
$$\overset{*}{\Rightarrow}a^iYb^{i+1}c^{i+1} \quad \text{(使用 } Xb\to bX,\ Xc\to Ybcc,\ bY\to Yb\text{)}$$
$$\Rightarrow a^{i+1}Xb^{i+1}c^{i+1}.$$

这里最后一步派生使用了规则 $aY\to aaX$. 这样就完成了归纳证明.

从 $a^iXb^ic^i(i\geqslant 1)$ 出发, 有

$$a^iXb^ic^i\overset{*}{\Rightarrow}a^iYb^{i+1}c^{i+1} \quad \text{(同上)}$$
$$\Rightarrow a^{i+1}b^{i+1}c^{i+1} \quad \text{(用 } aY\to aa\text{)},$$

其中 $i\geqslant 1$. 这就证明了 $L\subseteq L(G)$. 不难证明, 同时也有 $L=L(G)$, 这里从略.

关于上下文有关语言类 $\mathscr{L}(=\mathscr{L}(CS))$, 除了在§2.6提到的线性有界自动机问题未解决之外, 还有其他一些问题也没有解决. 最近, 一个著名问题, 即 $\mathscr{L}(CS)$ 关于补运算是否封闭已得到肯定性的答案.

§4 并行重写系统

前几节主要介绍了乔姆斯基的层次理论．这个理论在自然语言和计算机语言的研究中都是非常重要的，但它并不是在形式语言的复杂性研究中唯一可用的理论．在动力系统的语言复杂性研究中，我们还需要另一类很不一样的语言理论，即本节要介绍的 L 系统．它是由生物学家林登梅耶(A. Lindenmayer)建立的形式语言系统．本节材料的主要来源为[12]～[15]．

§4.1 最简单的 L 系统

回顾上一节中的生成语法 $G=(V, S, P, s_0)$，可见为了得到 L(G)中的字，要从初始符 s_0 出发，不断使用 P 中的语法规则进行重写(即改写)，直到符号串中不再含有 V 中的变量为止．这里的重写特征是每次使用一条规则，即一步派生(参见§3.1)．我们把这样的重写过程叫做**串行重写**．

林登梅耶在研究生物发育过程的数学模型时提出了与上述串行重写完全不同的**并行重写系统**．它的主要特征在于，在将一个符号串重写时，该符号串中的每一个符号都必须同时参与重写过程．由此建立起来的形式语言系统称为 **L 系统**．这里字母 L 是林登梅耶原姓的第一个字母．

实际上，L 系统包含了许多子系统，分别生成各类形式语言．这一小节先介绍最简单的 DOL 系统，其中 D 是指确定性，O 是指每一个符号在重写时与相邻符号无关．

DOL 系统是一个三元组

$$G=(S, h, \omega),$$

其中 S 为有限符号集，h 是从 S^* 映到 S^* 的同态映射，$\omega\in S^*$ 是初始符号串．由 G 生成的语言定义为

$$\mathrm{L}(G)=\{h^i(\omega)\mid i\geq 0\},$$

称为 DOL 语言.

关于同态, 在 § 1.9 的(6)已经遇到过. 一般而言, 设 S 和 Δ 是两个符号集, 如果映射 $h: S^* \to \Delta^*$ 满足条件 $h(\varepsilon)=\varepsilon$, 并且, 对 S^* 中的任何一对 x 和 y 成立 $h(xy)=h(x)h(y)$, 则称 h 为同态. 由此可见, 只要对每个符号 $a \in S$ 规定了 $h(a) \in \Delta^*$, 就给定了同态 h.

[例 1] 我们已经知道, $\mathrm{L}=\{0^{2^i} \mid i \geqslant 0\}$ 是上下文有关语言, 但不是上下文无关语言和正规语言(参见 § 1.7、§ 3.3 和 § 3.6).

现在定义一个 DOL 系统

$$G=(\{0\},\ \{0 \to 00\},\ 0).$$

可以直接看出 $\mathrm{L}=\mathrm{L}(G)$.

以后将说明, DOL 系统是 L 系统中最简单的一个子系统.

由这个例子已经可以看到, 形式语言的复杂性并没有一个绝对的度量. 本例中的语言 L 在乔姆斯基层次中为第 1 型语言, 复杂性层次相当高. 但它在 L 系统中则可以认为是非常简单的. 在下面还会看到相反的例子.

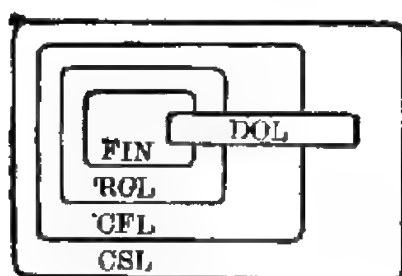


图 4-1

图 4-1 说明 DOL 语言与乔姆斯基语言层次的关系. 其中用 FIN 代表有限语言类, 即只含有限个符号串的语言全体. 因此从生成语言的能力来说, 可以看到, L 系统的理论也有自身的问题.

§ 4.2 OL、TOL 和 ETOL 系统

首先要去掉 DOL 系统中的确定性特征. 为此, 我们引入一般的置换, 来代替同态.

§ 1.9 已经提到过置换的概念. 现在作较为一般性的叙述. 设 S 和 Δ 是两个有限符号集. 对每个 $a \in S$, 设 $\sigma(a)$ 代表在 Δ 上的一个语言, 即 $\sigma(a) \subseteq \Delta^*$. 要求 $\sigma(\varepsilon)=\varepsilon$, 并且, 对在 S^* 中的任何两个串 w_1 和 w_2, 成立 $\sigma(w_1 w_2)=\sigma(w_1)\sigma(w_2)$. 这样, 就可以对于在

S 上的语言 L 定义

$$\sigma(\mathrm{L})=\{u \mid \text{对某个 } w\in \mathrm{L} \text{ 有 } u\in\sigma(w)\}.$$

我们称 L 为置换.

如果对每个 $a\in S$, $\sigma(a)$为有限集(即在 Δ 上的有限语言), 则称 σ 为有限置换. 如果对每个 $a\in S$, $\sigma(a)$只含一个符号串, 则 σ 就是 § 4.1 中介绍的同态. 特别, 当每个 $\sigma(a)\in\Delta$ 时, 称 σ 为编码.

现在介绍 OL 系统, 它是一个三元组

$$G=(S,\ \sigma,\ \omega),$$

其中的 S 和 ω 的意义与 DOL 系统相同, 所不同的是, 这里的 σ 是有限置换, 比同态更为广泛. 由 OL 系统 G 生成的语言为

$$\mathrm{L}(G)=\bigcup_{i\geqslant 0}\sigma^i(\omega),$$

称为 **OL 语言**.

[例 2] 我们知道 $\mathrm{L}=\{0^n \mid n\geqslant 0\}$ 是正规语言. 它的正规表达式即是 0^*. 可以证明: 这个 L 不是 DOL 语言, 即不能用任何 DOL 系统生成. 但只要取

$$G=(\{0\},\ \{0\to e,\ 0\to 00\},\ 0),$$

就得到 $\mathrm{L}=\mathrm{L}(G)$, 因此 L 是 OL 语言.

但是, 这样的 OL 系统在生成语言方面的能力仍不很强. 为此, 我们再介绍 TOL 系统.

TOL 系统也是一个三元组

$$G=(S,\ H,\ \omega),$$

其中的 S 和 ω 与过去一样, H 则是由有限个有限置换组成的非空集. 这就是说, H 的每一个元是一个有限置换. 对每个 $h\in H$, 三元组$(S,\ h,\ \omega)$是一个 OL 系统, 称为 G 的一个分系统. 由 G 生成的语言定义为

$$\mathrm{L}(G)=\{x\in S^* \mid x=\omega \text{ 或 } x\in h_1h_2\cdots h_k(\omega),\ \text{其中 } h_1,h_2\cdots,h_k\in H\}.$$

称 L(G)为 **TOL** 语言.

[例 3]　已知 $L=\{a^ib^ic^i \mid i\geqslant 0\}$ 是上下文有关语言，但不是上下文无关语言，当然更不是正规语言(参见 § 3.3 与 § 3.6).

现在令 $G=(S, H, \omega)$, 其中

$$S=\{a, b, c\},$$
$$\omega=abc,$$
$$H=\{h_1, h_2\},$$
$$h_1=\{a\to aa, b\to bb, c\to cc\},$$
$$h_2=\{a\to \varepsilon, b\to \varepsilon, c\to \varepsilon\}.$$

容易验证 $L=L(G)$. 因此, 这个 L 是一个 TOL 语言.

但是, 用 TOL 系统还不能生成所有的有限语言. 因此, 从这个角度来看, TOL 系统生成语言的能力还不够强. 这一点可从下面的例子得到了解.

[例 4]　$L=\{a, a^3\}$ 不是 TOL 语言, 当然也不是 OL 语言和 DOL 语言.

用反证法. 假设有一个 TOL 系统 $G=(S, H, \omega)$, 使 $L(G)=\{a, a^3\}$. 这时, 在 a 和 a^3 中总有一个是初始串 ω.

如果 $a=\omega$, 则从 a 出发经有限次并行重写后得到 a^3, 记为 $a\Rightarrow a^3$. 但由 a^3 出发也一定会有 $a^3\Rightarrow a^9$. 这里利用了应用重写规则时与相邻符号无关的特征. 这与 L 中只含两个字矛盾.

如果 $a^3=\omega$, 则从 $a^3\Rightarrow a$ 可见有 $a\Rightarrow\varepsilon$ 和 $a\Rightarrow a$ 成立. 但这样也就会有 $a^3\Rightarrow a^2$ 成立, 与 $a^2\notin L$ 矛盾.

于是, 我们看到, 一方面用最简单的 DOL 系统, 已经可以生成在乔姆斯基层次中比较复杂的某些上下文有关语言; 另一方面, 以上介绍的三种 L 系统还不能生成所有的有限语言, 而在乔姆斯基层次中有限语言是作为最简单的正规语言来看待的.

在 TOL 系统的基础上, 将生成语法系统中区分变量和终结符的做法借用过来, 情况就会改观. 这就是下面要介绍的 ETOL 系统.

ETOL 系统是一个四元组

$$G=(V,\ H,\ \omega,\ S),$$

其中 $U(G)=(V,\ H,\ \omega)$是一个 TOL 系统，称为 G 的基础系统，$S\subseteq V$ 是 G 的终结符号集. 由 G 生成的语言定义为

$$\mathrm{L}(G)=\mathrm{L}(U(G))\cap S^*.$$

换句话说，在符号集 V 中不属于 S 的符号起了变量的作用，而不允许在语言 L(G)的字中出现. 我们称 L(G)为 **ETOL** 语言.

完全相仿地，可以从 OL 系统出发定义 EOL 系统以及相应的 EOL 语言.

在 L 系统中，还有其他许多子系统，它们在生物学中都有特定的含义，这里不再一一介绍.

由于 L 系统具有并行重写的特征，近年来它在分形几何、植物生长模拟和计算机图形学方面得到应用. 对于这方面，可参看[14]、[15].

§4.3 语言类之间的关系

在 L 系统中，除了各种 OL(子)系统之外，还有各种 IL 系统. 在 IL 系统中的并行重写过程要将相邻符号的作用考虑进去. 与§3 相仿地，我们可以称 OL 系统为"上下文无关的 L 系统"，而将 IL 系统称为"上下文有关的 L 系统". 与 ETOL 系统的定义方式相仿，可以定义 EIL 系统以及相应的 EIL 语言.

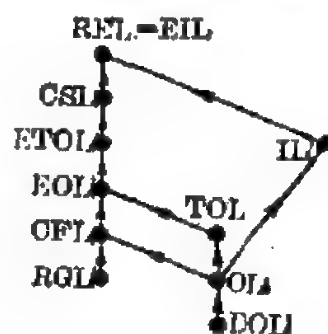


图 4-2

图 4-2 列出了以上介绍的几种 L 系统语言与乔姆斯基层次之间的关系. 其中，每个小圆点代表一个语言类. 记号 REL、CSL、CFL 和 RGL 代表乔姆斯基的四个层次. 从一个小圆点到另一个小圆点之间，如有有向线段，则表示较低的小圆点所代表的语言类为较高的小圆点所代表的语言类所真包含.

以后，用 $\mathscr{L}$(DOL)表示 DOL 语言全体所成的语言类．对于其他L系统也是如此．图 4-2 表明 $\mathscr{L}$(DOL)和 $\mathscr{L}$(OL)都是 $\mathscr{L}$(CS)的真子类，但与 $\mathscr{L}$(RG)或 $\mathscr{L}$(CF)则不存在包含关系(参见图 4-1)．对今后有用的包含关系是

$$\mathscr{L}(\mathrm{CF}) \subsetneqq \mathscr{L}(\mathrm{EOL}) \subsetneqq \mathscr{L}(\mathrm{ETOL}) \subsetneqq \mathscr{L}(\mathrm{CS}).$$

请注意，在L系统中最大的语言类 $\mathscr{L}$(EIL)与乔姆斯基层次中第0型语言类 $\mathscr{L}$(RE)相同．这是很有意义的事实．

§4.4 关于 ETOL 的一些性质

ETOL 系统是在上下文无关的L系统中最大的一个子系统，它在L系统理论中占有重要地位(参见[13])．

已经知道，语言类 $\mathscr{L}$(DOL)和 $\mathscr{L}$(OL)对许多简单运算都不封闭．与此不同的是，语言类 $\mathscr{L}$(ETOL)则具有良好的封闭性质，这在动力系统的应用中是重要的．

具体来说，$\mathscr{L}$(ETOL)对于以下六种运算都是封闭的：三种正规运算(即取和、连接和闭包)、同态、逆同态以及与正规语言取交(参见§1.9)．

在形式语言理论中，对以上六种运算封闭的语言类称为完全 **AFL** 类(Full Abstract Formal Languages)．已经知道，$\mathscr{L}$(RG)、$\mathscr{L}$(CF)、$\mathscr{L}$(RE)和 $\mathscr{L}$(ETOL)都是完全 AFL 类．但是，$\mathscr{L}$(CS)不是完全 AFL 类．

成为对比的是，$\mathscr{L}$(OL)对上述六种运算的每一种都不封闭．$\mathscr{L}$(EOL)则对其中的同态和逆同态不封闭．

可以证明，存在一个单调增长的语言类序列 $\{\mathscr{L}_n\}_{n>0}$，其中每一个 $\mathscr{L}_n$ 都是完全 AFL 类，使成立

$$\mathscr{L}(\mathrm{ETOL}) = \mathscr{L}_0 \subsetneqq \mathscr{L}_1 \subsetneqq \cdots \subsetneqq \mathscr{L}_n \subsetneqq \cdots \subsetneqq \mathscr{L}(\mathrm{CS}).$$

注意这里的记号 $\mathscr{L}_n$ 与乔姆斯基层次的记号无关．

作为一个例子，我们来证明图 4-2 中的

$$\mathscr{L}(\mathrm{CF}) \subsetneqq \mathscr{L}(\mathrm{EOL}).$$

这同时也建立了 $\mathscr{L}(\mathrm{CF})\subsetneqq\mathscr{L}(\mathrm{ETOL})$。

从§4.1的例1已知 $\mathscr{L}(\mathrm{CF})\neq\mathscr{L}(\mathrm{EOL})$，以下只要证包含关系成立.

设 $\mathrm{L}\in\mathscr{L}(\mathrm{CF})$。按照定义，存在一个第2型语法 $G=(V, S, P, s_0)$，使 $\mathrm{L}=\mathrm{L}(G)$.

现定义一个OL系统

$$G'=(V\cup S,\ P\cup\{a\to a\mid a\in V\cup S\},\ s_0).$$

容易验证

$$\mathrm{L}=\mathrm{L}(G')\cap S^*.$$

实际上，从 G' 的构造可见在 G 中的每一步派生可看成为在 G' 中的并行重写. 证毕.

§4.5 标号语言

作为上下文无关语言的推广，又具有并行重写特征的另一类语言是标号语言(Indexed Languages)。它在动力系统中也有应用. 关于标号语言，可以阅读阿霍(A. V. Aho)的原文[16]. 下面只介绍其定义，并举一个例子.

标号语言的语法是第2型语法的一种扩充. 它是一个五元组

$$G=(V, S, I, P, s_0).$$

与§3.1相比，多了一个标号集 I. 记 A、$B\in V$ 为变量，$\alpha\in(V\cup S)^*$ 为符号串，$f\in I$ 为标号，则在 P 中的语法规则有三种类型：

1° $A\to\alpha$,

2° $A\to Bf$,

3° $Af\to\alpha$.

由此可见，标号或标号串可以在变量后面出现，但不能接在终结符后面. 换句话说，派生过程中的每个表达式都是 $(VI^*\cup S)^*$ 中的符号串.

设 β、$\gamma\in(VI^*\cup S)^*$, $\delta\in I^*$, $X_i\in V\cup S$, $i=1, 2, \cdots, k$, 则在标号语言中的三类语法规则的使用方式如下：

1° 对于 $A\to\alpha$, 记 $\alpha=X_1X_2\cdots X_k$, 则有

$$\beta A\delta\gamma\Rightarrow\beta X_1\delta_1X_2\delta_2\cdots X_k\delta_k\gamma,$$

其中，当 $X_i\in V$ 时 $\delta_i=\delta$，当 $X_i\in S$ 时 $\delta_i=\varepsilon$. 这就是将标号串 δ 并行地分配到 α 中的每一个变量后面去.

2° 对于 $A\to Bf$, 简单地有

$$\beta A\delta\gamma\Rightarrow\beta Bf\delta\gamma.$$

3° 对于 $Af\to\alpha=X_1X_2\cdots X_k$, 有

$$\beta Af\delta\gamma\Rightarrow\beta X_1\delta_1X_2\delta_2\cdots X_k\delta_k\gamma,$$

其中关于 δ_i 的规定与 1° 中相同.

可将以上看成为标号(串)的分配、增加和使用(或消耗). 在 1° 和 3° 中都具有并行重写的特征.

定义语言

$$\mathrm{L}(G)=\{w\mid s_0\overset{*}{\Rightarrow}w \text{ 且 } w\in S^*\}$$

为 G 生成的标号语言.

[例 5] $\mathrm{L}=\{a^ib^ic^i\mid i\geqslant1\}$是标号语言(可与 § 4.2 例 3 对比).

令

$$G=(\{s_0, A, B, C, T\}, \{a, b, c\}, \{f, g\}, P, s_0),$$

其中有两个标号 f 和 g, 语法规则为

$$\begin{aligned}P=\{&s_0\to Tg,\ T\to Tf,\ T\to ABC,\\&Af\to aA,\ Bf\to bB,\ Cf\to cC,\\&Ag\to a,\ Bg\to b,\ Cg\to c\}.\end{aligned}$$

容易验证以下派生过程:

$$\begin{aligned}s_0&\overset{*}{\Rightarrow}Tf^{i-1}g\\&\Rightarrow Af^{i-1}gBf^{i-1}gCf^{i-1}g\Rightarrow a^ib^ic^i,\end{aligned}$$

其中 $i\geqslant1$. 这就不难证明 $\mathrm{L}=\mathrm{L}(G)$.

记标号语言类为 $\mathscr{L}(\mathrm{IND})$.

从定义与例 5 已经得到

$$\mathscr{L}(\mathrm{CF})\subsetneqq\mathscr{L}(\mathrm{IND}).$$

在[16]中证明了

$$\mathscr{L}(\mathrm{IND})\subsetneqq\mathscr{L}(\mathrm{CS}).$$

语言类 $\mathscr{L}(\mathrm{IND})$ 具有良好的封闭性，它是一个完全 AFL 类。

在[17]中指出

$$\mathscr{L}(\mathrm{ETOL})\subsetneqq\mathscr{L}(\mathrm{IND}),$$

因此，其他 OL 系统生成的语言类也都是 $\mathscr{L}(\mathrm{IND})$ 的真子类.

由上可知，$\mathscr{L}(\mathrm{IND})$ 在图 4-2 中的位置，恰好在代表 $\mathscr{L}(\mathrm{CS})$ 的圆点和代表 $\mathscr{L}(\mathrm{ETOL})$ 的圆点之间的直线段上。

第2章

区间映射与形式语言

本章以单峰映射[†]为例，讨论在区间映射动力系统中自然产生的形式语言. 在§5介绍符号动力学[††]的基本概念. 以此为基础，在§6中给出形式语言的定义. 我们将从几种不同的角度来讨论这个定义本身的合理性. 这里的主要结果是，这样定义的形式语言恰好反映了动力系统在任意有限时间内的符号动力学行为. 相应定理(即§6定理1)的证明放在附录A中. 此外，还证明了在形式语言中出现的周期符号序列必定是动力系统中某个真实周期轨的反映.

本章为后面的研究作准备.

§5 区间映射的符号动力学

本节叙述符号动力学方面的基本概念，为引进形式语言作准备. 其中有关结论大多数不作证明. 请读者参看所列举的参考文献.

§5.1 单峰映射

设 f 是在区间 $I=[0, 1]$ 上有定义并映入到 I 的映射. 如果满足以下条件:

1° f 连续;

2° f 在区间 I 的某个内点 c 处达到极大值 $f(c)$, 在 $[0, c)$

† 参看本套丛书中郝柏林著《从抛物线谈起——混沌动力学引论》一书.

†† 参看本套丛书中郑伟谋、郝柏林著《实用符号动力学》一书.

上严格单调增加，在$(c, 1]$上严格单调减少；

3° $f(0)=f(1)=0$,

则称 f 为单峰映射，并称点 c 为 f 的临界点.

在单峰映射中，研究得最多的是二次方映射（族），另一个常用名称是洛吉斯蒂映射(Logistic Maps). 这是一族带有参数的单峰映射：

$$f(x)=bx(1-x),$$

其中 $0<x<1$，参数 b 的范围为 $0\leqslant b\leqslant 4$. 只要取 $I=[0, 1]$, $b\neq 0$, f 就满足上述三个条件. 在文献中出现的二次方映射还有 $1-ax^2$, $a-x^2$ 等等形式. 容易证明，在一定的意义下它们都是等价的.

作为函数来考虑，单峰映射中的 f 并不复杂. 特别是二次方映射中的 f，就是中学里已熟知的一元二次函数，它在几何上是一条抛物线. 但作为动力系统来说，单峰映射则表现出极其丰富和复杂的性态，许多重要问题只是在不久前才刚解决，或者迄今尚未解决. 现在讨论更为一般的区间映射. 从映射生成动力系统的主要过程是迭代. 从区间 I 的某个点 x 出发，用 f 不断作迭代，得到序列 x, $f(x)$, $f(f(x))$, $\cdots$. 采用记号 $f^2(x)=f(f(x))$, 一般地记 $f^n(x)=f(f^{n-1}(x))$, $n\geqslant 1$，就可以将这个序列记为 $\{f^n(x)\}_{n\geqslant 0}$. 其中约定 $f^0(x)=x$. 我们又可以将它记为

$$f^*(x)=(x, f(x), f^2(x), \cdots, f^n(x), \cdots),$$

称为"在迭代映射 f 下从点 x 出发的轨（或轨道）". 这样，就从一个映射 f 生成一个动力系统. 我们可将迭代次数 n 理解为离散化了的时间的第 n 个时刻. 于是 x 的轨 $f^*(x)$ 就代表动力学行为不断演化的记录.

问题的复杂性在于，即使对于像二次方映射这样简单的情况，除了个别参数值之外，我们也不能建立轨道 $f^*(x)$ 的解析表达式. 为了知道 n 充分大时轨道 $f^*(x)$ 的动力学行为，必须寻求新的方法.

70年代以来，关于区间迭代映射的研究已有极为重要的进展. 诸如混沌、奇怪吸引子等重要的动力系统概念的形成和发展，都与区间迭代映射的研究有密切关系. 以下列举作者所见到的部分文献，供读者参考.

李天岩(Li. T. Y.)和约克(J. A. Yorke)在有关区间映射的论文[18]中，第一次在科学意义上使用了混沌(Chaos)这个名词，虽然其中的一部分结果只是沙可夫斯基(A. N. Sharkovskii)的工作[19]的特例. 梅(R. M. May)的综述[20]对于区间迭代映射的研究起了重要的推动作用，其中也对早期的研究作了介绍. 米尔诺(J. Milnor)和萨斯顿(W. Thurston)的长篇文章[21]是在区间迭代映射方面的基本文献，它以预印本的形式自1976年以来已广为流传. 关于区间迭代映射的第一本专著，是库列(P. Collet)和埃克曼(J. -P. Eckmann)的[22]， 它将在本书中多次被引用. 其他部分或全部涉及区间迭代映射的专著有[23]～[29]等.

§5.2 符号动力学

先引进一些基本概念.

最简单的轨道是**不动点**，即 $f(x)=x$ 的点 x. 它的轨是常值序列. 其次是**周期轨**，即存在某个自然数 p，使得 $f^p(x)=x$. 这时，对任何 $n\geqslant 0$ 有 $f^{n+p}(x)=f^n(x)$ 成立，轨 $f^*(x)$ 是周期序列. 我们称 p 为**周期**，称轨中的点为**周期点**或者**周期 p 点**. 如果 p 是具有上述性质的最小自然数，即使得 x, $f(x)$, $\cdots$, $f^{p-1}(x)$ 各不相同，但成立 $f^p(x)=x$，则称 p 为这个周期轨的**最小周期**. 不动点可认为是最小周期为1的周期轨.

如果在轨 $f^*(x)$ 的序列中从某项开始是周期 p 序列，则称轨 $f^*(x)$ 为**终极周期轨**(Eventually Periodic Orbit). 相应地，可以定义终极周期轨的周期和最小周期. 当然，可以将周期轨看成终极周期轨的特殊情况.

以上是三种最简单的轨．在区间迭代映射的动力系统研究中的一个重要成就，在于发现极为简单的区间映射，在作为动力系统来研究时，会表现出极其复杂的动力学行为．为了研究它们的复杂性，已经发展出许多新的概念、理论和方法，例如符号动力学、重正化群、李雅普诺夫指数、测度熵、拓扑熵、分维等等．

下面介绍符号动力学的有关内容，并且以此为基础开展对区间迭代映射的语言复杂性研究．

对于单峰映射，我们约定，当点落在$[0, c)$中时用符号0来记，而当点落在$(c, 1]$中时用符号1来记，对于临界点就用c来记，不再引入新的符号．这样，就可以将轨$f^*(x)$转化为由符号0、1和c组成的符号序列．注意这时的符号0、1并不代表数值，改用其他符号也是可以的．

将以上做法形式化．先定义将$x\in[0, 1]$转化为符号的函数

$$A(x)=\begin{cases}0, & \text{当 } x\in[0, c)\text{ 时；}\\ c, & \text{当 } x=c\text{ 时；}\\ 1, & \text{当 } x\in(c, 1]\text{ 时．}\end{cases}$$

然后定义

$$A(f^*(x))=(A(x),\ A(f(x)),\ \cdots,\ A(f^n(x)),\ \cdots).$$

我们称这个序列为"初始点x的踪迹(Itinerary)"，并记为$I(x)$ $(=A(f^*(x)))$，以强调它对x的直接依赖关系(我们也称$I(x)$是轨$f^*(x)$的踪迹)．

用$I(x)$代替轨$f^*(x)$，这就是粗粒化的描述方法．初看起来这似乎太粗糙，势必丧失许多信息，但从符号动力学目前已取得的丰富成果来看，这种方法是非常成功的．这方面除了文献[18]，[19]，[21]～[29]之外，还可以参看莫尔斯(M. Morse)等较早期的研究工作[30]～[34]．

今后，我们将动力系统在符号序列中反映出来的动态行为称为"符号动力学行为"．它与系统的真实轨道的动力学行为之间的关系将在下面论及．

对于将单峰映射经过迭代而生成的动力系统来说，非常根本的一个事实是：几乎所有符号动力学行为都是由临界点 c 的踪迹 $I(c)(=A(f^i(c)))$所决定的．去掉这个序列的第一个符号 c，即是踪迹 $I(f(c))$，特称为"单峰映射 f 的揉序列(Kneading Sequence)"．

今后经常用 KS(f)表示单峰映射 f 的揉序列，在不发生混淆时，简记为 KS．

§5.3 符号序列之间的序

为了研究任何一点 $x\in I$ 的踪迹 $I(x)$与揉序列 KS(f)之间的关系，我们需要在符号序列之间引入"序"，即某种大小关系．

虽然每个踪迹都是由三个符号 0，1 和 c 组成的无限符号序列，但用这三个符号任意组成的序列不一定是某个点 $x\in I$ 的踪迹．实际上，由符号 c 与临界点的对应关系可以看出，如果在一个踪迹 $I(x)$的某一位上出现 c，那么在这个符号 c 之后只能重复揉序列 KS(f)．又，如果在 KS(f)中出现 c，则这显然说明临界点 c 是一个周期点 c．从而这时在 KS(f)将无穷多次出现 c，KS(f)本身成为周期序列．与此相同的是，如果任何踪迹 $I(x)$中出现符号 c，也就会无穷多次出现符号 c(往往只写到第一个 c)．

上述关于符号 c 出现的特征也就成为一个符号序列是某个点的踪迹的必要条件；当然，这并不是充分条件．

设 s 是由符号 0、1 和 c 组成的一个无限序列，记为

$$s=s_1s_2\cdots s_n\cdots.$$

并设 s 已满足上述关于符号 c 出现时的必要条件．在所有这样的无限序列所成的集合上，定义一个**移位算子** σ 为：

$$\sigma(s_1s_2\cdots)=s_2s_3\cdots.$$

如果 s 为某点 x 的踪迹 $I(x)$，即 $s=I(x)$，则容易直接验证有

$$\sigma(I(x))=I(f(x)).$$

这一性质也可以记为 $\sigma\circ I=I\circ f$．

在上述序列之间按以下方式引入序(即大小关系)，并借用记号 <、>、⩽、⩾ 和 = 表达小于、大于等关系.

先在三个基本符号之间规定序为

$$0<c<1.$$

注意，这并非是关于数的不等式.

其次，对于只由符号 0 和 1 组成的有限符号序列(即符号串)，定义奇串和偶串如下：在符号序列中如符号 1 出现奇数次，则为奇串，出现偶数次，则为偶串.

现设 $s=s_1s_2\cdots$ 和 $t=t_1t_2\cdots$ 是由符号 0、1 和 c 组成的无限序列，且均满足关于出现符号 c 的必要条件. 比较 s 和 t. 如果对每个 $i\geqslant 1$ 都有 $s_i=t_i$，则定义 $s=t$；否则，即有 $s\neq t$. 这时，从 $i=1$ 开始，比较 s_i 和 t_i：如果 $s_1\neq t_1$，则定义 $s<t$(或 $s>t$)$\Leftrightarrow s_1<t_1$($s_1>t_1$)；一般地，设有 $s_1=t_1$，…，$s_n=t_n$，但 $s_{n+1}\neq t_{n+1}$(其中 $n\geqslant 1$). 这时称 $s_1\cdots s_n(=t_1\cdots t_n)$为 s 和 t 的公共主部. 因为 s 和 t 满足关于符号 c 出现的必要条件，所以在公共主部中不会出现符号 c(否则不会发生 $s_{n+1}\neq t_{n+1}$)，即完全由符号 0 和 1 组成. 我们规定，当 $s_1\cdots s_n(=t_1\cdots t_n)$为偶串时，

$$s<t(s>t)\Leftrightarrow s_{n+1}<t_{n+1}(s_{n+1}>t_{n+1}).$$

而当 $s_1\cdots s_n$ 为奇串时，

$$s<t(s>t)\Leftrightarrow s_{n+1}>t_{n+1}(s_{n+1}<t_{n+1}).$$

然后，按自然方式定义记号 ⩽ 和 ⩾ 的含义.

这样引进的序的合理性，可以从下列事实得到说明，即从 x 到 $I(x)$ 的保序性质：

$$x、y\in I,\ x<y\Rightarrow I(x)\leqslant I(y).$$

关于这个事实的证明和对于序的进一步了解，可参见[22]或已提到的其他文献.

§5.4 必要条件和充分条件

现设 s 是由 0、1 和 c 组成的无限序列，并在出现符号 c 时满

足在§5.3一开始所说的条件. 关于 s 是否为某点 $x\in I$ 的踪迹,有以下基本结论:

1° 如 $s=I(x)$ 对某点 $x\in I$ 成立,则对每个 $i\geqslant 1$,成立

$$\sigma^i(s)\leqslant \mathrm{KS}.$$

2° 如果 KS 不含符号 c,且对每个 $i\geqslant 1$,成立

$$\sigma^i(s)<\mathrm{KS},$$

那么存在点 $x\in I$,使 $s=I(x)$.

这里的条件1°是 s 为踪迹的必要条件,而条件2°则是 s 为踪迹的充分条件. 对于 KS 中含有符号 c 时的充分条件,这里不再给出. 下面对此另有说明.

必要条件1°的证明很容易. 因为 $f(c)$ 是单峰映射 f 在区间 I 上的最大值,所以对任何点 $x\in I$ 和任何 $i\geqslant 1$,总有 $f^i(x)\leqslant f(c)$ 成立. 利用§5.3末提到的保序性质,就得到

$$I(f^i(x))\leqslant I(f(c))=\mathrm{KS}.$$

再利用在引入移位算子 σ 时所指出的性质 $\sigma\circ I=I\circ f$,上式左方即等于 $\sigma^i(I(x))=\sigma^i(s)$. 证毕.

充分条件2°的证明要困难一些,请参看[21]、[22].

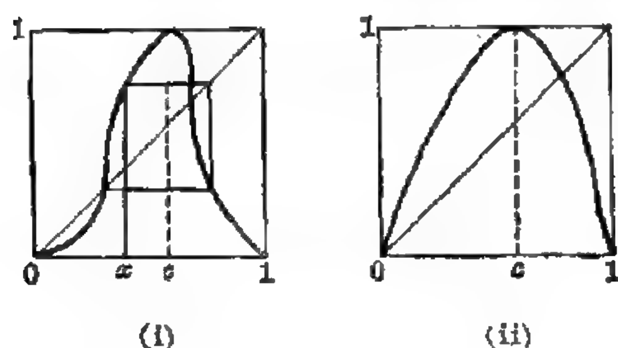


图 5-1

由于必要条件1°与充分条件2°并不相同,自然会提出问题: $s=I(x)$ 的充分必要条件是什么? 这不是一个容易回答的问题. 实际上,这与具体的映射有关,不可能作出包罗所有情况的回答. 在图5-1中给出了两个单峰映射的几何图象,它们的揉序列都是 10^∞.

观察图5-1(i)中点 x 的踪迹,易见 $I(x)=010^\infty$. 如考虑点 x 的某些原象,则不难找到以 0^n10^∞ 为踪迹的点,其中 n 可以为任何一个自然数.

但图 5-1(ii)的情况完全不同. 对任何自然数 n, 不存在点 $x\in I$, 使 $I(x)=0^n10^\infty$ 利用在(ii)中 $x=0$ 是在$[0, c)$中唯一的不稳定不动点，这一点是容易证明的.

现在检验 $s=0^n10^\infty(n>0)$ 与条件 1° 和 2° 的关系. 从 § 5.3 中的序的定义容易看出，对每个 $i\geq 1$ 有 $\sigma^i(s)\leqslant KS=10^\infty$. 但当 $i=n$ 时，这里成立等号，因此条件 2° 不满足.

从以上讨论可见，图 5-1(i)说明了充分条件 2° 并非是必要的，$s=0^n10^\infty(n>0)$不满足条件 2°，但仍存在 x 使 $s=I(x)$. 另一方面，图 5-1(ii)则说明必要条件 1° 并不是充分的.

最后指出，从必要条件推出揉序列本身必是移位最大字，即 $\sigma^i(KS)\leqslant KS$ 对每个 $i\geq 0$ 成立. 可以证明这也是成为揉序列的充分条件.

§ 6 形式语言的定义

在符号动力学的基础上，本节给出单峰映射的形式语言定义，并对其合理性进行探讨. 可以看出，这种定义形式语言的方法有其必然性.

§ 6.1 从允许字定义形式语言

首先，我们不准备考虑含有符号 c 的踪迹. 实际上，从单峰映射的定义可以看出，任何一个点的原象不超过两个. 因此，临界点 c 的所有(有限次)原象用数学语言来说最多只有可列个. 从这种原象点出发的轨在未到达临界点 c 之前，与邻近点的轨的符号动力学行为是相同的. 而在到达临界点 c 之后的符号动力学行为即是简单地重复揉序列 KS. 因此，去掉这些原象，对于动力系统的符号动力学行为并没有丢失什么. 这样在下面我们就只研究仅仅由符号 0 和 1 组成的踪迹.

其次，我们引进允许字的概念，它只与给定的揉序列有关，而

与具体的映射无关.

设已给定一个揉序列 KS. 我们称由符号 0 和 1 组成的无限序列 s 为关于 KS 的一个允许字, 如果对每个 $i\geqslant 0$ 满足条件

$$\sigma^i(s)\leqslant \mathrm{KS}.$$

这个定义与 § 5.4 中的必要条件 1 只有一点差别, 即在 $i=0$ 时要求不等式也成立. 容易看出, 任何点 $x\in[0, f(c)]$, 只要它不是 c 的某次原象, 它的踪迹 $I(x)$ 就是符合上述定义的允许字. 另一方面, 当 $x\in(f(c), 1]$ 时, $I(x)$ 不符合上述条件, 因此不是允许字. 然而, 由于这时必有 $f(x)<f(c)$ 成立, 因此 $I(f(x))$ 仍然是允许字; 只要 $f(x)$ 不是 c 的某次原象.

由以上讨论可见, 引进允许字的概念相当于把讨论范围从区间 $I=[0, 1]$ 缩小到 $[0, f(c)]$ 上. 由于单峰映射生成的动力系统的主要动力学行为总是发生在 $[0, f(c)]$ 中, 因此这是可以接受的. 对于由此引起的形式语言定义上的差异将在下面讨论(见 § 6.6).

留下的两个问题是: 允许字未必反映真实的踪迹(这在图 5-1 的讨论中已经见到), 同时允许字为无限符号序列, 不能应用现成的形式语言工具来研究.

采取以下做法可以同时解决这两个问题. 即只要取所有允许字的所有有限子串构成一个形式语言. 它将具有下面说明的良好性质.

这里要指出, 当 s 为允许字时, 对任何 $i\geqslant 0$, $\sigma^i(s)$ 也是允许字. 因此从所有允许字取所有有限子串的做法, 与从所有允许字取所有前缀(Prefix)的做法是完全等价的(这里要使用在 § 1.2 中介绍的一些概念).

现在引进正式的数学记号. 设 f 为一个单峰映射, KS(f)为揉序列. 定义形式语言

$$\mathrm{L}=\{t\in\{0, 1\}^* \mid \text{存在允许字 } s\text{, 使 } t \text{ 为 } s \text{ 的前缀}\}.$$

我们称 L 是"单峰映射 f 的形式语言", 也称 L 是"由揉序列

KS(f)确定的形式语言".

由允许字的定义可见，L 实际上只由揉序列确定，而与具体的映射 f 无关. 因此更为方便的记法是引进作用在揉序列 KS 上的算子 $\mathscr{L}$，并记

$$L=\mathscr{L}(KS).$$

如上所述，算子 $\mathscr{L}$ 的含义是，从 KS 出发确定所有允许字，然后取它们的所有前缀.

我们总是要求 $s\in L$. 这里 s 为空串. 由于 s 是任何符号序列的前缀，因此可以认为 $s\in L$ 已隐含在 L 的定义之中.

我们发现，对于语言 L 来说，在讨论图 5-1 时那样的困难并不存在. 这就是下面的一个定理的主要意义.

定理 1 设 f 为单峰映射，KS 为 f 的揉序列，语言 $L=\mathscr{L}(KS)$，那么对任何字 $t\in L$，存在点 $x\in[0, f(c)]$，使 t 是踪迹 $I(x)$ 的前缀.

这个定理的证明见附录 A. 我们只在这里解释一下它的意义. 以图 5-1(ii)为例. 这时 $KS=10^{\infty}$. 对任何 $n>0$，0^n10^{∞} 为符合定义的允许字. 我们已经知道，它们不代表任何点 x 的真实符号动力学行为. 取它的前缀，得到三种类型的符号串：

$$0^i(1\leqslant i\leqslant n),\qquad 0^n1,\qquad 0^n10^m(m>0).$$

不难从图 1-5(ii)上直接找出点 x，使 $I(x)$ 以这些符号串中的任何一个为前缀. 这样就表明，虽然不存在 $x\in I$，使 $I(x)=0^n10^{\infty}$，但这个允许字的任何一个前缀则反映了真实的符号动力学行为，无论这个前缀的长度如何.

从 KS 出发计算 $\mathscr{L}$(KS)的具体例子将在 §7 中举出.

本节以下部分用于讨论定义 $L=\mathscr{L}(KS)$ 中的各种理论问题.

§6.2 揉序列含符号 c 的情况

在语言 $L=\mathscr{L}(KS)$ 的定义中，揉序列 KS 可以含有符号 c，但允许字和 L 中的字则只由符号 0 和 1 组成. 当 KS 含 c 时只写

到 c 也够了.

在今后我们主要考虑不含符号 c 的揉序列. 我们在这里说明一下,这样做对于语言 $L=\mathscr{L}(KS)$的讨论不会造成损失.

实际上,如果有一个揉序列

$$KS=a_1a_2\cdots a_{n-1}c\cdots,$$

且设于串 $a_1a_2\cdots a_{n-1}$ 中不出现 c. 由 § 5.3 的讨论可见这时的 KS 为周期序列:

$$KS=(a_1a_2\cdots a_{n-1}c)^{\infty}.$$

现在规定, 当 $a_1a_2\cdots a_{n-1}$ 为偶串时取 $a_n=0$, 而当 $a_1a_2\cdots a_{n-1}$ 为奇串时取 $a_n=1$(关于偶串和奇串的定义见 § 5.3), 然后定义

$$KS_1=(a_1a_2\cdots a_n)^{\infty}$$

利用[22]、[26]中的结果, 可以知道 KS_1 也是某个单峰映射的揉序列.

从允许字的定义容易知道,如 s 为关于 KS 的允许字,则它也是关于 KS_1 的允许字,反之亦成立. 这样就得到

$$\mathscr{L}(KS)=\mathscr{L}(KS_1).$$

因此在今后可以限于只讨论不含符号 c 的揉序列所确定的形式语言.

顺便指出, 上述处理方法与[21]对类似情况所用的方法完全相同.

§ 6.3 周期允许字与周期轨

今后可以看出, 从揉序列出发容易确定各种周期允许字的存在性. 但是有一个基本问题要讨论, 即周期允许字的存在与原来的单峰映射的真实周期轨的存在之间有什么关系. 对于图 5-1 的讨论已说明一个允许字未必是真实轨的反映. 可以证明, 对于周期允许字, 这种情况不会发生. 换句话说, 在讨论周期轨的问题上,粗粒化描述方法是完全成功的. 确切的结论如下.

定理 2 设 KS 是不含符号 c 的揉序列. 如果 $s=B^{\infty}$ 是关于

KS 的一个周期允许字，最小周期为 $|B|$ (即串 B 的长度)，那么存在周期点 $x \in I$，使 $s = I(x)$，同时周期轨 $f^*(x)$ 的最小周期与 s 的最小周期相同．如果 B 为奇串，则还可能存在最小周期为 $2|B|$ 的周期轨．

现在给出这个定理的证明．

不妨设 $s = B^\infty$ 已是移位最大字，即对每个 $i \geqslant 0$ 成立 $\sigma^i(s) \leqslant s$．否则，可以在

$$s, \ \sigma(s), \ \cdots, \ \sigma^{|B|-1}(s)$$

中取最大者代替 s，以符合这个假设．

先证明存在某个点 $x \in I$，使 $s = I(x)$．如果 $\mathrm{KS} = s$，则已有 $s = I(f(c))$．如果 $\mathrm{KS} \neq s$，则有 $s < \mathrm{KS}$．由于 s 是移位最大字，因此对每个 $i \geqslant 0$，成立

$$\sigma^i(s) < \mathrm{KS}.$$

应用§5.4中的充分条件，就知道存在 $x \in I$，使 $s = I(x)$．但这个 x 未必为周期点．

现在引用符号动力学方面的一个基本定理．它的内容是：如果某个踪迹 $I(x) = AB^\infty$ 为终极周期序列，其中 B 的长度已为最小，则轨 $f^*(x)$ 收敛于某个周期轨．这个周期轨的最小周期当 B 为偶串时等于 $|B|$，当 B 为奇串时等于 $|B|$ 或者 $2|B|$．关于这个定理的证明可看[21]、[22]等文献．

应用这个定理于 $s = I(x)$，当 B 为偶串时得到周期为 $|B|$ 的周期轨，当 B 为奇串时从定理的证明可看出也存在周期为 $2|B|$ 的周期轨(但未必为极限周期轨)．

利用临界点 c 不是周期点的这个条件，可以从定理的证明中推出上述周期为 $|B|$ 的周期轨的踪迹即是 B^∞．当 B 为奇串时还可能存在倍周期轨，它也以 B^∞ 为踪迹．证毕．

反之，如果已有一个周期轨 $f^*(x)$，那么它的踪迹 $I(x) = A(f^*(x))$ 是周期序列．将上述证明中引用的定理直接用到这个周期轨 $f^*(x)$ 上，可见 $I(x)$ 的最小周期或与轨 $f^*(x)$ 的最小周期

相等，或者是它的一半.

对于揉序列中含符号 c 的情况，定理 2 需要修改. 这时存在某些周期允许字，它们不是任何点的踪迹. 由于下面主要讨论不含 c 的揉序列，我们不再讨论这个问题.

§6.4 由语言确定揉序列

我们已经定义了由揉序列 KS 确定的形式语言 $\mathscr{L}(\mathrm{KS})$. 在某些情况下，我们需要从语言出发研究揉序列. 这时下面的结果是有用的.

设 KS_1 和 KS_2 是两个单峰映射的揉序列，它们都不含有符号 c，那么有

$$\mathscr{L}(\mathrm{KS}_1)=\mathscr{L}(\mathrm{KS}_2)\Rightarrow \mathrm{KS}_1=\mathrm{KS}_2.$$

证明是简单的. 从 KS_1 来看，它本身是移位最大字，因此它也是关于自身的允许字. 由语言 $\mathscr{L}(\mathrm{KS}_1)$ 的定义可见，KS_1 的每一个前缀都是语言 $\mathscr{L}(\mathrm{KS}_1)$ 中的字.

从 $\mathscr{L}(\mathrm{KS}_1)=\mathscr{L}(\mathrm{KS}_2)$，可见 KS_1 的每个前缀也是 $\mathscr{L}(\mathrm{KS}_2)$ 中的字. 由 $\mathscr{L}(\mathrm{KS}_2)$ 的定义推出 KS_1 的每个前缀 $\leqslant \mathrm{KS}_2$. 由此即知 $\mathrm{KS}_1\leqslant \mathrm{KS}_2$.

同样可以证明 $\mathrm{KS}_2\leqslant \mathrm{KS}_1$，因此 $\mathrm{KS}_1=\mathrm{KS}_2$.

这个结果表明，不同的揉序列所确定的形式语言一定是不相同的. 这就是说，从揉序列 KS 到形式语言 $\mathrm{L}=\mathscr{L}(\mathrm{KS})$ 的对应关系是一一对应.

§6.5 语言定义的修改

研究从揉序列定义形式语言的其他方法.

可以认为在 §6.1 中的定义方法是以 §5.4 中的必要条件为基础的. 如改成以那里的充分条件为基础，可定义另一形式语言：

$$\mathrm{L}_1=\{t\in\{0,1\}^*\mid \text{存在由符号 0 和 1 组成的无限序列 } s,\ s \text{ 以 } t \text{ 为前缀，且对每个 } i\geqslant 0 \text{ 满足条件 } \sigma^i(s)<\mathrm{KS}\},$$

从定义可知，

$$L_1 \subseteq L = \mathscr{L}(KS).$$

容易找出例子，使 $L_1 \neq L$. 这方面有结果，

定理 3 如果揉序列 KS 不是周期序列，则有

$$L_1 = L = \mathscr{L}(KS).$$

这个结果不仅探讨了形式语言定义的第二种可能方式，在今后研究非周期揉序列时也有用处。定理 3 的证明亦写在附录 A 中。

§6.6 语言定义的另一种修改

对于 $L = \mathscr{L}(KS)$ 还可以作另一种修改. 如果将 §6.1 中允许字的定义改成只要求对 $i \geq 1$ 满足 $\sigma^i(s) \prec KS$, 那么所得到的形式语言将不仅反映了 $x \in [0, f(c)]$ 时的动力学行为，而且也包括了 $x \in (f(c), 1]$ 时的动力学行为. 我们将这样的语言记为 $\bar{L}$.

今后将看到，对于 L 进行研究比较方便. 同时，如 §6.1 中所说，将区间 $I = [0, 1]$ 改为 $[0, f(c)]$ 不减少讨论的一般性.

但是 $\bar{L}$ 与 L 的关系仍应考虑. 特别是在作语言复杂性分析时，讨论 $\bar{L}$ 和 L 是否等价，这是一个需要解决的问题.

从定义即可建立

$$\bar{L} = \{0, 1\}L,$$

其中将 $\{0, 1\}$ 看成只由两个长度为 1 的字组成的语言，等式右方是两个语言的连接(参见 §1.9).

也可以将 $\bar{L}$ 与 L 的关系改写为

$$\bar{L} = \{0\}L \cup \{1\}L.$$

现在可以建立以下结果.

如果考虑在第 1 章中介绍的以下六个语言类中的任何一个：$\mathscr{L}(RG)$、$\mathscr{L}(CF)$、$\mathscr{L}(CS)$、$\mathscr{L}(RE)$、$\mathscr{L}(ETOL)$ 和 $\mathscr{L}(IND)$，则 L 和 $\bar{L}$ 一定同属于其中的一类.

这就是说，在上述意义下，对 L 和 $\bar{L}$ 进行语言复杂性分析是等价的.

以下对这个结果作出证明.

记 $\mathscr{L}$ 为其中的任何一个语言类.

由于这六个语言类关于连接和取并运算封闭，因此从 $\mathrm{L}\in\mathscr{L}\Rightarrow\mathbf{L}\in\mathscr{L}$.

反之，设 $\mathbf{L}\in\mathscr{L}$. 利用 $\mathscr{L}$ 关于正规语言取交为封闭，可推出

$$\{0\}\mathrm{L}=\mathbf{L}\cap\{0\}(0+1)^{*}\in\mathscr{L}.$$

定义从 $\{a, 0, 1\}^{*}$ 到 $\{0, 1\}^{*}$ 的一个同态为

$$h(a)=0,\qquad h(0)=0,\qquad h(1)=1,$$

其中 a 为一个辅助符号. 这时有

$$\{a\}\mathrm{L}=h^{-1}(\{0\}\mathrm{L})\cap\{a\}(0+1)^{*}.$$

利用$\mathscr{L}$关于逆同态封闭，可见 $\{a\}\mathrm{L}\in\mathscr{L}$.

再引进一个同态 g 为

$$g(a)=\varepsilon,\qquad g(0)=0,\qquad g(1)=1,$$

并利用 $\mathscr{L}$ 关于有限删除运算(Limited Erasing)封闭，就得到所求的结论：

$$\mathrm{L}=g(\{a\}\mathrm{L})\in\mathscr{L}.$$

以上所用到的知识可参看[1]的第 11 章.

第 3 章

区间映射中的正规语言

上一章给出了由揉序列确定的语言 $\mathscr{L}$(KS)的定义. 这一章将对于其中的正规语言进行彻底的分析. 主要结果是给出了从揉序列 KS 出发直接判定 $\mathscr{L}$(KS)为正规语言的方法, 同时在 $\mathscr{L}$(KS)为正规语言时, 给出了从 KS 出发构造接受 $\mathscr{L}$(KS)的最小确定性有限自动机的方法.

第 7 节对于语言 $\mathscr{L}$(KS)所具有的两个基本性质作出讨论, 对于给定符号串 z 和揉序列 KS 建立 $z \in \mathscr{L}$(KS)的判定方法. 同时,引进一个符号串相对于某一揉序列的前后缀这个重要概念. 它在建立有关理论时是非常有用的工具.

第 8 节以几个定理的形式给出本章的主要结果. 在研究接受 $\mathscr{L}$(KS)的有限自动机的基础上, 证明 $\mathscr{L}$(KS)为正规语言时, KS 一定是周期序列或终极周期序列. 它的逆命题也成立, 其证明则请看附录 B. 在那里还建立了一些重要的概念和引理. 在这节中还以 KS$=100(101)^\infty$为例介绍了马尔可夫划分 (Markov Partition)的方法.

在第 9 节中对于以上两类正规语言给出了相应的有限自动机的构造方法, 介绍了最小确定性有限自动机的状态个数的计算方法, 列举了大量的例子以供参考.

在一定条件下可以将揉序列为周期序列时的动力学行为看成为正规运动, 而将揉序列为终极周期序列时的动力学行为看成为随机性很强的粗粒混沌. 本章的结果表明, 从形式语言的角度来看, 这两种系统的复杂程度都不高, 在乔姆斯基层次中处于最低的正规语言这个层次上(关于它们之间的差别将在第 15 节中讨论).

应当着重指出的是，对上述两种运动的复杂性刻划，采用形式语言工具得到了其他已有方法所没有提供的新的认识.

§7 关于语言的一般性讨论

本节先举一些例子，然后指出语言 $L=\mathscr{L}(KS)$ 所具有的两个基本性质. 它们也是本书中所涉及的来自动力系统的其他形式语言所共同具有的性质. 对于给定的符号串 x 和揉序列 KS，给定了 $x\in\mathscr{L}(KS)$ 的判定方法. 引入前后缀概念为今后作准备.

本节材料主要取自[35]～[39]，所需要的语言知识见本书的§1.

§7.1 关于满射情况的讨论

作为 $L=\mathscr{L}(KS)$ 的第一个例子，研究单峰映射 f: $[0, 1]\to[0, 1]$ 为满射的情况. 按照§5.1中的定义，这时的揉序列为 10^∞. 这也就是图5-1中的情况. 对于二次方映射族，即参数 $b=4$ 的情况：

$$f(x)=4x(1-x),\qquad 0\leqslant x\leqslant 1.$$

[例1] 对于 $KS=10^\infty$，求 $L=\mathscr{L}(KS)$.

按照§5.3中关于序的定义，10^∞ 是最大可能的符号串. 因此，由符号0和1组成的任何无限序列 s 都满足允许字的条件：对每个 $i\geqslant 0$ 成立 $\sigma^i(s)\leqslant 10^\infty$. 由此可知，语言 $\mathscr{L}(10^\infty)$ 中含有由符号0和1组成的任何有限序列，即

$$\mathscr{L}(10^\infty)=S^*,$$

其中 $S=\{0, 1\}$.

由此可见，$\mathscr{L}(10^\infty)$ 是正规语言，从所包含的字的多少来看，是单峰映射中最大可能的形式语言. 关于这个语言的最小确定性有限自动机及其正规表达式可参看第1节中的图1-2. 从以后的叙述会知道，在区间映射的形式语言中，没有比这个例子更为简单

的情况.

将这一点与其他方法的研究结果作比较是很有意义的. 下面所提到的各种概念可参见[21]～[29]或本丛书中其他各册的介绍.

李雅普诺夫指数(Lyapunov Exponent)刻划动力系统的未来性态对初值的微扰动的依赖关系，即关于初值的敏感性. 对于二次方映射的满射情况，即 $f(x)=4x(1-x)$，可以用解析方式计算出李雅普诺夫指数 $\lambda=1$（在计算中取对数的底为 2)，即对初值的敏感性很强. 但容易看出，李雅普诺夫指数不可能仅仅从揉序列来确定，而必须依赖于映射的进一步的分析性质. 这与我们从揉序列 10^∞ 出发的研究角度是不一样的.

关于测度熵的情况大致相同. 测度熵是关于某个不变测度来计算的量，而不变测度的存在及其性质与映射的分析性质有密切联系. 这里不再讨论.

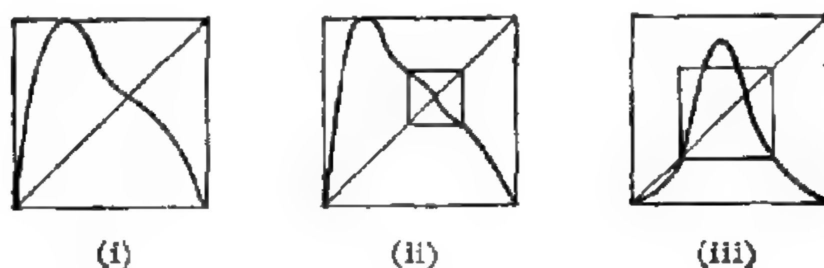
拓扑熵的研究与形式语言方法有密切联系. 第 5 章将从语言中字的多样性角度对拓扑熵作详细研究.

从 $\mathscr{L}(10^\infty)=S^*$ 可以看出，这里所反映的纯粹是轨道的符号动力学行为的多样性. 这时的多样性达到了如此程度，以致于任何一个 $\mathscr{L}(\mathrm{KS})$中的字在 $\mathscr{L}(10^\infty)$中都出现. 换句话说，任何一个单峰映射的任何轨的符号动力学表现，即§5.2中的 $I(x)=A(f^n(x))$，在满射单峰映射的动力系统中都能找到，只要取适当的初始点 x 即可. 其中包括具有任何周期行为的周期轨.

上面用形式语言方法证明了这一点仅仅由揉序列 $\mathrm{KS}=10^\infty$ 所决定，而与映射的其他性质无关. 可以将这种性质看成为某种普适性或通用性. 形式语言方法表明，正由于这种普适性或通用性，即多样性达到最大程度，因此其语言结构却是简单语言(指正规语言)中最简单的一个.

实际上这完全由 $\mathrm{KS}=10^\infty$ 决定，而与映射 f 是否是满射，是否有稳定周期轨，是否具有某种确定类型的非游荡集都没有关系.

图 7-1 列举以 10^∞ 为揉序列的三个单峰映射，作为对图 5-1 的补充. 其中(i)有稳定不动点，(ii)有稳定周期 2 轨，(iii)以一个康托集(Cantor Set)和一个不动点为非游荡集. 它们在 $\mathscr{L}(10^\infty)=S^*$ 上完全一致. 我们可以说，如果符号动力学是通过粗粒化描述来研究动力系统的方法，则采用形式语言为工具是一个很自然的发展.



(i)　　(ii)　　(iii)

图 7-1

§ 7.2 两个简单例子

[例 2] 对于 $KS=0^\infty$ 求形式语言.

与 10^∞ 恰恰相反，$KS=0^\infty$ 是所有揉序列中最小的一个. 由此已可推出，满足 § 6.1 定义的允许字只有一个，即 $KS=0^\infty$ 自身. 这样就求出

$$\mathscr{L}(0^\infty)=\{0^n|n\geqslant 0\}.$$

它是正规语言，其正规表达式为 0^*(参见 § 1.5). 在图 7-2 中给出了以 0^∞ 为揉序列的两个单峰映射和接受语言 0^* 的最小确定性有限自动机. 这个自动机只有两个状态. 初态也就是终止态，用一个小圆点外加一个圆圈代表. 另一个是非终止态，由一个空心小圆表示.

图(i)中的映射以 $x=0$ 为稳定不动点. 图(ii)中的映射以 $x=0$ 为不稳定不动点，而在区间$(0, c)$中还有一个稳定不动点. 当然还可以作出更复杂的例子，包括有无限多个不动点的情况. 但由于唯一的允许字为 0^∞，从 § 6.3 的定理 2，可知这时不可能有不动点之外的任何周期轨.

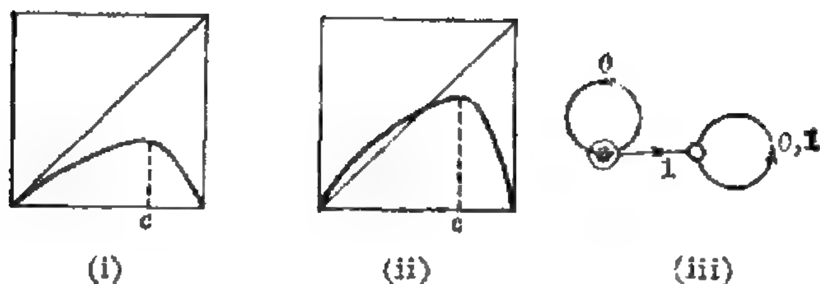


图 7-2

对于二次方映射 $f(x)=bx(1-x)$，这相当于 $0<b<2$ 的情况.

[例 3] 对于 $KS=1^{\infty}$ 求形式语言.

按照允许字的定义可见只有 0^{∞} 和形状为 $0^n1^{\infty}(n\geqslant 0)$ 这样的允许字. 取它们的所有前缀,就得到形式语言

$$\mathscr{L}(1^{\infty})=\{0^n1^m|n\geqslant 0,\ m\geqslant 0\}.$$

这即是在第 1 章的图 1-3 中讨论过的正规语言，它的正规表达式是 0^*1^*. 在图 1-3(ii)作出了接受 $\mathscr{L}(1^{\infty})$的最小确定性有限自动机.

从 $KS=1^{\infty}$ 容易看出只存在两个周期允许字, 即 0^{∞} 与 1^{∞}. 按照 § 6.3 的定理 2, 可知存在以 0^{∞} 为踪迹的不动点, 以 1^{∞} 为踪迹的不动点, 还可能有以 1^{∞} 为踪迹的周期 2 轨.

以二次方映射为例, 参数 b 处于$(2,\ 1+\sqrt{5})$时揉序列为 $KS=1^{\infty}$. 其中 $b\in(2,\ 3]$时只有不动点, $b\in(3,\ 1+\sqrt{5})$时还有周期 2 轨.

图 7-3 是以 1^{∞} 为揉序列的一个单峰映射的图象和接受语言 $\mathscr{L}(1^{\infty})$的最小确定性有限自动机.

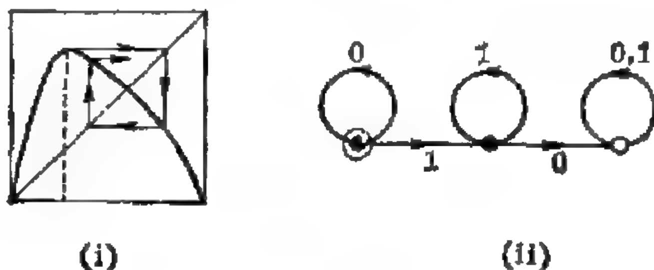


图 7-3

§7.3 关于正规语言的一般问题

现在考虑一般问题：揉序列 KS 具有什么性质时，语言 $\mathscr{L}$(KS)是正规语言？已经有一系列工作指出，当揉序列为周期序列或终极周期序列时，单峰映射的符号动力学行为可以用正规语言描述(参见[35]～[49])。反之，在[35]、[36]、[38]中则证明，当 $\mathscr{L}$(KS)为正规语言时，揉序列 KS 只可能是上述两种情况．因此可以认为，在单峰映射的形式语言中，正规语言的问题已经解决.

在第 8 节将从揉序列 KS 出发直接研究语言 $\mathscr{L}$(KS)的性质，而不必如上面几个例子那样每次要求出全部允许字．在第 9 节将对于 $\mathscr{L}$(KS)为正规语言的情况，给出构造相应的最小确定性有限自动机的具体算法．这样我们就完全掌握了 $\mathscr{L}$(KS)的性质，并提供了种种正规语言之间复杂程度的比较依据.

这里采取在[38]中的方法，以自然等价关系 R_L 和迈希尔-奈罗德定理为主要工具(参见 §1.8)，同时还需要发展一些新的概念和方法．在本节的以下部分为此作准备.

§7.4 $\mathscr{L}$(KS) 的两个基本性质

对于任何揉序列 KS，语言 $\mathscr{L}$(KS)都具有以下所指出的两个基本性质。

设符号集为 S，语言 $\mathrm{L}\subseteq S^*$.

性质 1 $z\in \mathrm{L} \Leftrightarrow z$ 的每一个子串 $z'\in \mathrm{L}$.

性质 2 $z\in \mathrm{L} \Rightarrow$ 至少存在一个 $a\in S$，使符号串 za 也属于 L.

现对这两个性质作些解释.

由于 z 自身也是 z 的一个子串，因此性质 1 中的“$\Leftarrow$”是平凡的，重要的是其另一半“$\Rightarrow$”．语言的许多进一步的性质均与此有关．性质 1 是语言 $\mathrm{L}=\mathscr{L}$(KS)的定义的直接推论．它具有明显

的动力学意义. 联系到第6节的定理1,性质1表明,如果某种符号动力学行为能够出现,则它的任何局部行为也一定能够出现.

另一方面,不与动力系统对应的形式语言,就不一定具有这个性质. 如在§1.7、§3.3中所列举的形式语言例子都不具有性质1.

因此,性质1看起来平常,实际上对于动力系统中的形式语言来说是非常重要的基本特征. 除了区间映射之外,本书第6章元胞自动机中出现的形式语言也具有性质1.

性质2也不是一般的形式语言所具有的公共特征,但对于动力系统中的形式语言却是经常具有的性质. 对于 $L=\mathscr{L}(KS)$, 如 $z\in L$, 则根据定义(§6.1),存在一个允许字 s, 以 z 为 s 的前缀.由于 s 是无限符号序列,可见性质2成立.

同样可以看出,性质2具有明显的动力系统意义,即时间演化在理论上不受限制,从当前时刻总要演化到下一时刻.

值得指出,莫尔斯和海德隆特(G. A. Hedlund)在用符号动力学方法研究负曲率曲面上的测地线时,曾经对于所使用的符号串集合(即这里所说的形式语言)指出过与上述两个性质相类似的几个特征(参见[32]的832页)

§7.5 $z\in\mathscr{L}(KS)$ 的判定法则

在这一小节中要解决的问题是,对于给定的符号串 z 和揉序列 KS, 判定 $z\in\mathscr{L}(KS)$ 是否成立. 注意在§7.4中的性质1并未给出这个问题的直接回答. 按照§6.1中语言 $\mathscr{L}(KS)$ 的定义,要找出一个无限符号序列 s, 它以 z 为前缀,又满足对每个 $i\geqslant 0$ 的条件 $\sigma^i(s)\leqslant KS$, 这也是不切合实际的.

$z\in\mathscr{L}(KS)$ 的判定法则是,条件

$$\sigma^i(z)\leqslant KS$$

对 $0\leqslant i<|z|$ 的每个 i 成立. 这里是将有限串与 KS 作比较,等号成立的意义是指左方的有限串为右方的揉序列的前缀.

由于 $\sigma^i(z)$ 在 $0\leqslant i<|z|$ 时是 z 这个符号串的所有非空后缀，包括 z 自身，因此这个判定方法的应用是很方便的.

从语言 $L=\mathscr{L}(KS)$ 的定义容易看出，任何一个有限符号串，如按 § 5.3 的序的定义不小于等于KS，这个符号串一定不属于 L.

另一方面，z 的每个后缀当然也是 z 的子串. 从性质 1 可见，$z\in L\Rightarrow z$ 的每个后缀属于 L. 综合以上，可见在判定方法中给出的条件是 $z\in L$ 的必要条件. 这里重要的是证明这个条件同时也是充分条件. 这个证明将在下面 § 7.7 中给出.

这里要指出，如一个符号串 z 的所有前缀，包括 z 本身在内，都满足小于等于 KS 的相同条件，仍不足以保证 $z\in L$. 例如，设

$$KS=1011011010110\cdots,$$

$$z=10110110100,$$

z 的每一个前缀都满足上述条件. 但是 z 有一个后缀为 100. 从 $100>KS$，可见 $100\notin\mathscr{L}(KS)$. 根据性质 1，也可知 $z\notin\mathscr{L}(KS)$.

这里使用了性质 1 的逆否形式，即当 z 有一个子串 $z'\notin L$ 时，即可推出 $z\notin L$. 这是性质 1 经常使用的形式.

§ 7.6 符号串的前后缀

在证明上一小节的判定方法时，需要引进一个符号串关于揉序列的前后缀概念. 从已进行的研究来看，这个概念非常有用.

给定 $z\in S^*$ 和揉序列 KS，称非空串 z' 是 z 关于 KS 的前后缀(Prefix-Suffix)，如果 z' 满足下列条件:

1° z' 是 KS 的前缀，

2° z' 是 z 的后缀.

在不发生混淆时，简称上述 z' 是 z 的一个前后缀.

类似地可以定义 z 关于某一个符号串的前后缀.

如果 z 有关于 KS 的前后缀，则可以将这样的前后缀中最长的一个定义为 z 关于 KS 的最长前后缀(The Longest Prefix-Suffix). 在许多论证中它起重要作用.

现在讨论前后缀的存在性.

如果 z 本身是 KS 的一个前缀，则 z 本身即是 z 关于 KS 的(最长)前后缀. 这是平凡情况.

如 $z\notin\mathscr{L}(\mathrm{KS})$，则 z 未必有关于 KS 的前后缀. 在上一小节中的例子即是如此.

如 $z\in\mathscr{L}(\mathrm{KS})$，$z$ 仍然可以没有关于 KS 的前后缀. 例如

$$z=01, \quad \mathrm{KS}=0^{\infty},$$

或者

$$z=000, \quad \mathrm{KS}=(10)^{\infty},$$

都是如此. 但实际上这是 $z\in\mathscr{L}(\mathrm{KS})$ 时仅有的两种例外情况. 这里有下面的结果.

设非空串 $z\in\mathscr{L}(\mathrm{KS})$，其中 $\mathrm{KS}\neq 0^{\infty}$，$z$ 不是只由符号 0 组成的串，那么 z 存在关于 KS 的前后缀，从而也存在关于 KS 的最长前后缀.

证明 由于 KS 是移位最大字，既然它不会是 0^{∞}，则 KS 必从符号 1 开始.

如果 z 的最后一个符号是 1，则 $z'=1$ 就是 z 关于 KS 的一个前后缀.

如果 z 的最后一个符号为 0，则由于 z 不是 0 串，因此 z 的后缀中一定有一个具有形式为 10^m 的后缀，$m>0$. 由于 $z\in\mathrm{L}$，从性质 1 可见 $10^m\in\mathrm{L}$，因此 $10^m\leqslant\mathrm{KS}$. 这保证了 KS 一定以 10^m 为前缀. 因而 10^m 是 z 关于 KS 的前后缀.

§7.7 判定法则的证明

如前所述，对于给定的 z 和 KS，只要证明，如 z 的每一个后缀(包括 z 自身)$z'\leqslant\mathrm{KS}$，则 $z\in\mathrm{L}=\mathscr{L}(\mathrm{KS})$.

按照 $\mathscr{L}(\mathrm{KS})$ 的定义，需要构造一个允许字 s，它以 z 为前缀.

分两种情况. 如 z 的每一个后缀 z' 满足条件 $z'<\mathrm{KS}$，则只要令

$$s = z \cdot \mathrm{KS},$$

其中记号"·"为符号串的连接. 这时对每个 $i \geqslant 0$ 成立

$$\sigma^i(s) \prec \mathrm{KS}.$$

另一种情况是 z 的某些后缀为 KS 的前缀，即在 $z' \preccurlyeq \mathrm{KS}$ 中成立等号的情况. 这时 z 有关于 KS 的前后缀. 取其中最长的一个，即 z 关于 KS 的最长前后缀，将 z 分解为

$$z = z_1 z_2,$$

其中 z_2 是 z 关于 KS 的最长前后缀.

现在取

$$s = z_1 \cdot \mathrm{KS},$$

这个无限符号序列 s 以 z 为前缀(因 KS 以 z_2 为前缀). 考虑对 $i \geqslant 0$ 的每个 $\sigma^i(s)$.

如果 $i \geqslant |z_1|$，则

$$\sigma^i(s) = \sigma^{i-|z_1|}(\mathrm{KS}) \preccurlyeq \mathrm{KS}.$$

这里利用了 KS 为移位最大字；

如果 $i < |z_1|$，则

$$\sigma^i(s) = \sigma^i(z) \cdot \mathrm{KS}.$$

由于 z_2 是 z 关于 KS 的最长前后缀，而目前

$$|\sigma^i(z)| > |z_2|,$$

因此成立 $\sigma^i(z) < \mathrm{KS}$，从而也有

$$\sigma^i(s) < \mathrm{KS}.$$

这样我们就证明了 s 为允许字，从而得到 $z \in \mathrm{L}$.

一个简单推论：$z \in \mathrm{L} = \mathscr{L}(\mathrm{KS})$ 的充分必要条件是，z 的每一个子串 $z' \preccurlyeq \mathrm{KS}$.

§8 从揉序列判定正规性

本节讨论如何从揉序列 KS 直接判定 $\mathscr{L}(\mathrm{KS})$ 是否为正规语

言，并给出充分必要条件．材料主要取自[35]、[36]、[38]等．定理2和3的证明放在附录B中．

§8.1 有限自动机的特征分析

从§7.4知道语言$L=\mathscr{L}(KS)$必然具有的两个基本性质．作为它们的应用，我们来研究当L为正规语言时，接受L的有限自动机必然会具有的一些特征．

这里关心的问题是，从给定的揉序列出发．直接研究语言以及自动机的性质．主要的工具是在§1.8介绍的自然等价关系R_L和迈希尔-奈罗德定理．为读者方便起见，以下先重复叙述R_L的定义．

设$L\subseteq S^*$为一个形式语言，$S=\{0, 1\}^*$，则L在S^*中产生一个自然等价关系R_L：称S^*中的串x和y为R_L等价，记为xR_Ly，如果对任何串$z\in S^*$，xz和yz同时属于L，或者同时不属于L．容易验证R_L满足等价关系的三个条件(见§1.8)．将R_L等价类记为$[x]$，其中x是这个类中的任何一个符号串．这样的记法就是说以x作为这个等价类的代表元．

称等价类个数为R_L的指标．迈希尔-奈罗德定理的一个内容是，证明L为正规语言的充分必要条件是R_L为有限指标．这个定理的另一个内容是，将每一个等价类作为一个状态，就可以从等价类之间的自然关系构造出接受语言L的自动机．如果R_L为有限指标，这就是接受正规语言L的最小确定性有限自动机．如果R_L为无限指标，则所得到的是具有无限个状态的自动机．它除了状态为无限这一点之外，在其他方面与确定性有限自动机完全一样．在本节中的自动机即指这两类自动机．

从R_L的定义可见，如xR_Ly，取$z=e$(即空串)，则推出x和y或同属于L，或都不属于L．这是xR_Ly的必要条件．

从$L=\mathscr{L}(KS)$的基本性质1(见§7.4)可以看出有

$$x\notin L,\ y\notin L\Rightarrow xR_Ly \text{ 成立．}$$

这是因为对任何 $z \in S^*$，串 xz 和 yz 都不会在语言 L 中．

在本书中用记号 L′ 表示语言 $L \subseteq S^*$ 在 S^* 中的补，即

$$L' = S^* - L,$$

称 L′ 为 L 的补语言．

由上可见，对于具有性质 1 的任何形式语言 L，它的补 L′ 必是 R_L 的一个等价类，即有

$$x, y \in L' \Rightarrow xR_L y,$$

$$x \in L', yR_L x \Rightarrow y \in L'.$$

注意空串 $\varepsilon \in L$ 乃是性质 1 的推论．因此总有 $\varepsilon \notin L'$ 成立．

联系到迈希尔-奈罗德定理的证明过程(见[1])，我们知道在用 R_L 等价类构造自动机时，如 $x \in L$，则$[x]$为自动机中的终止状态，如 $x \notin L$，则$[x]$为自动机的非终止状态．因此得到结论，接受具有性质 1 的形式语言的自动机，只需要一个非终止状态．

同样还可以知道，由以上非终止状态出发的每条弧只能转移到非终止状态．图 7-2 和 7-3 上的自动机都具有这些特征．

为方便起见，我们约定在今后的自动机示意图中，略去从非终止状态(它用空心小圆代表)出发的弧．实际上这种状态即所谓死状态，一旦进入，再不出来．

唯一的例外是 $KS = 10^\infty$，这时 $L = \mathscr{L}(10^\infty) = S^*$，$L' = \varnothing$，因此不存在非终止状态(见图 1-2)．

现在考虑 §7.4 中的基本性质 2．它在自动机中的意义是非常清楚的，即从任何一个终止状态出发，在标有 0 和 1 的弧中间至少有一条通向到一个终止状态．也就是说，不可能发生从一个终止状态出发的所有弧都通向到非终止状态的情况．

换一个说法，如果 L 具有性质 2，M 为接受它的自动机，那么可以在 M 中将非终止态以及通向它的所有弧都去掉，这时从留下的任何一个终止状态出发，在状态转移图中存在长度为任意的路径．

§ 8.2 计算 R_L 等价类的例子

设 $L=\mathscr{L}(KS)$，记 R_L 的指标为 N

［例 1］ $N=1 \Leftrightarrow KS=10^{\infty}$.

这里的"$\Leftarrow$"部分已于 § 7.1 得到；实际上从 $L=S^*$ 可知 R_L 只有一个等价类．它的自动机见图 1-2.

现在考虑"$\Rightarrow$"的证明.

如果 $L'=\varnothing$，则由于每一个形状为 10^m 的串属于 L，利用 $10^m \leqslant KS$ 对每个 m 成立即可推出 $KS=10^{\infty}$.

如果 $L'\neq\varnothing$，则由于 $L=\mathscr{L}(KS)$ 也不会是空的，因此与 R_L 的指标 $N=1$ 相矛盾．这时至少有两个 R_L 等价类.

［例 2］ $N=2 \Leftrightarrow KS=0^{\infty}$.

讨论必要性部分（即"$\Rightarrow$"）．如果 $N=2$，从例 1 可见 $L\neq S^*$，即 $L'\neq\varnothing$．于是已经有一个 R_L 等价类为 L'．由 $\varepsilon\in L$ 可知有一个等价类 $[\varepsilon]\subset L$.

现在分析 $[\varepsilon]$ 中的内容，这时容易确定对任何 $m\geqslant 0$ 有 $0^m R_L \varepsilon$ 成立．实际上如果 KS 不是 0^{∞}，就必是从符号 1 开始．如果 $z\in L$，则按照 § 7.4 的判定法则，考虑 $0^m z$ 的所有后缀，即可推出 $0^m z\in L$．反之，$0^m z\in L \Rightarrow z\in L$ 是明显的.

现在观察在等价类 $[\varepsilon]$ 中是否含有其他符号串．如果假设在 $[\varepsilon]$ 中含有一个不全为符号 0 组成的串，则可看出符号 1 作为字已属于 L（应用性质 1）．因此揉序列 KS 从 1 开始．由于 $N=2$，我们也有 $1\in[\varepsilon]$ 成立.

从 $0^m R_L 1$ 出发，取 $z=0^n$，n 任意，就推出 $0^{m+n} R_L 10^n$，特别是由此得出 $10^n\in L$ 对每个 $n\geqslant 0$ 成立．这直接导致 $KS=10^{\infty}$．引用例 1，可见与 $N=2$ 相矛盾.

这样就从 $N=2$ 推出 $1\in L'$，KS 从 0 开始，因此 $KS=0^{\infty}$.

这时的两个等价类为

$$L' \text{ 与 } [\varepsilon]=\{0^m \mid m\geqslant 0\}.$$

从 $KS=0^{\infty}$ 出发，容易确定 $L=\mathscr{L}(0^{\infty})$，这在§7.2的例2中已讨论过，这里从略.

[例3]　$N=3 \Leftrightarrow KS=1^{\infty}$.

设 $N=3$ 成立，则从例2可知揉序列KS必是从符号1开始的.

利用在例2中的同样讨论可以知道这时存在 R_L 的三个等价类：$[\varepsilon]$，$[1]$ 与 L'．其中 $[\varepsilon]=\{0^m|m\geqslant 0\}$ 与例2也相同.

考虑符号串10．这时有两种可能．如果 $10\in L$，则KS也必定从10开始．从 $N=3$ 和例1又知 $KS\neq 10^{\infty}$．设 $KS=10^n1\cdots$，$n>0$．由于 $N=3$，从 $10\in L$ 只能有 $10\in[1]$，即 $10R_L1$．取 $z=0^n$，则 $10z\notin L$，而 $1z\in L$，引出矛盾.

于是只可能是 $10\in L'$，同时KS不能以10为前缀．因此就得出 $KS=1^{\infty}$ 的结论.

从 $KS=1^{\infty}$ 推出 $N=3$ 是比较容易的，这里从略．关于 $\mathscr{L}(1^{\infty})$ 的讨论见§7.2的例3.

注：对于 R_L 的指标 $N>3$ 的情况，由以上讨论已知有三个等价类：$[\varepsilon]$、$[1]$ 与 L'．其中 $[\varepsilon]=\{0^m|m\geqslant 0\}$ 总是成立的.

§8.3　主要结果及其证明

现在叙述并证明本节的主要结果.

定理1　如果由揉序列KS确定的形式语言 $\mathscr{L}(KS)$ 为正规语言，则KS是周期序列或终极周期序列.

证明：记语言 $L=\mathscr{L}(KS)$．由于L是正规语言，存在接受L的最小确定性有限自动机 M，记为

$$M=(Q,\ S,\ \delta,\ q_0,\ F).$$

其中 $S=\{0,\ 1\}$，Q 为有限状态集，$F\subseteq Q$ 为终止状态集，$q_0\in Q$ 为初态，δ 为转移函数．关于这些概念，参见本书§1或在那里介绍的文献.

从§8.1的讨论已知，除了 $KS=10^{\infty}$ 这个唯一的例外，在

$Q-F$ 中只含有一个状态，它对应于 R_L 的等价类 L′. 由于 KS $=10^\infty$ 已是终极周期序列，在 § 8.2 的例 1 中已作过讨论，因此在证明的以下部分假定 KS$\neq 10^\infty$. 这时 L$\neq S^*$, L′$\neq \varnothing$.

在 § 8.1 中已经证明，从集合 F 的任何一个状态出发，都存在长度为无限的路径，而且它完全在集合 F(以及有关的弧的集合)之中.

从 $s\in$ L 可见初态 $q_0\in F$.

现在从 q_0 出发寻找完全落在 F 中的一条特殊的路径，它在今后被称为极大路径.

这时利用在 § 1.4 中介绍的状态转移图从几何上观察问题是非常方便的，这时状态即是结点.

记一条无限长路径的对应符号序列为

$$t=t_1t_2\cdots t_n\cdots,$$

每个 t_i 是相应弧的标号，采用以下方式归纳地确定 t_i.

从代表 q_0 的结点出发，有两条标以 0 和 1 的弧. 如果其中有一条通向非终止状态 L′，则将另一条弧的标号取为 t_1. 我们知道不可能发生两条弧都通向 L′ 的情况(见 § 8.1). 如果两条弧都通向到 F 中的状态，则取 $t_1=1$.

现设已经确定了从 q_0 出发长度为 i 的路径，它对应的符号序列为 $t_1t_2\cdots t_i$，并将这时所到达的终止状态记为 q_i. 按照以下规则确定 t_{i+1} 以及相应的弧：

1° 从 q_i 出发的两条弧都通向 F 中的状态. 这时要计算

$$\max\{t_1\cdots t_i1,\ t_1\cdots t_i0\},$$

然后取其中较大的串的最后一个符号为 t_{i+1}.

2° 从 q_i 出发只有一条弧通向 F 中的状态. 这时就取这一条弧的标号为 t_{i+1}.

由 § 8.1 中的讨论知道，不存在从 q_i 出发的两条弧都通向非终止状态 L′ 的情况.

这样我们就得到了所要的长度无限的路径，以及相应的无限

符号序列 t. 称此路径为极大路径. 根据§6.2的讨论,可以假定 KS 不含 o. 这时我们证明 $t=\mathrm{KS}$. 首先，从 M 为确定性的自动机可知，任何一个有限符号串 $s\in S^*$ 对应于从 q_0 出发的一条完全确定的路径,又从 L 具有§7.4中的性质1可知,如果 $s\in \mathrm{L}$, 则 s 所对应的路径完全落在 F 中. 将 s 与 t 作比较，利用 t 的构造过程,可见 $s\leqslant t$ 成立.

由于可取 KS 的每个前缀为 $s(\in \mathrm{L})$，因此已证明了 $\mathrm{KS}\leqslant t$. 另一方面，由于 t 完全在 F 中，t 的每个前缀是自动机所接受的字，因此知道 t 的每个前缀都小于等于 KS, 即也有 $t\leqslant \mathrm{KS}$ 成立. 于是 $t=\mathrm{KS}$ 成立.

证明的最后部分是要证明上述序列 t 是终极周期序列. 当然，t 也可能是周期序列,因为它可以看成是终极周期序列的特殊情况.

写出在生成序列 t 的过程中的结点(状态)与弧的序列:

$$q_0 \xrightarrow{t_1} q_1 \xrightarrow{t_2} q_2 \xrightarrow{t_3} \cdots \xrightarrow{t_i} q_i \xrightarrow{t_{i+1}} q_{i+1} \rightarrow \cdots .$$

由于每个 $q_i\in F$, 而 F 为有限集，因此在 $\{q_i\}_{i\geqslant 0}$ 中必有重复. 这里要分两种情况讨论.

情况1 在 $\{q_i\}_{i\geqslant 0}$ 中存在无限多个结点具有性质：从它们出发的两条弧都通向 F 中的状态. 考虑相应的 $(q_i, t_{i+1})\in F\times S$, 则一定能找到 j 与 l, 使

$$q_j = q_{j+l}, \quad t_{j+1} = t_{j+l+1},$$

其中 $l>0$.

由于从 q_j 出发的两条弧都通向 F 中的状态，因此按照 t 的生成规则可得到

$$t_1\cdots t_j t_{j+1} > t_1\cdots t_j \bar{t}_{j+1}$$

和

$$t_1\cdots t_j t_{j+1}\cdots t_{j+l} t_{j+l+1} > t_1\cdots t_j t_{j+1}\cdots t_{j+l} \bar{t}_{j+l+1},$$

其中 $\bar{t}_{j+1}$ 是 t_{j+1} 的反码，$\bar{t}_{j+l+1}$ 是 t_{j+l+1} 的反码. 这里反码按照 $\bar{0}=1$, $\bar{1}=0$ 来理解.

由此可见 $t_{j+1}\cdots t_{j+l}$ 是偶串，即为含偶数个 1 的串．不难用数学归纳法证明，对每个 $k\geqslant 0$ 成立

$$q_{j+k}=q_{j+l+k},\quad t_{j+1+k}=t_{j+l+1+k}.$$

这就导致所要的结论：

$$\mathrm{KS}=t=t_1\cdots t_j(t_{j+1}\cdots t_{j+l})^{\infty}.$$

情况 2 对于 $\{q_i\}_{i\geq 0}$ 存在 N．当 $j>N$ 时，每个 q_j 只有一条弧通向 F 中的状态．这时容易直接看出存在 $P>0$，使

$$\mathrm{KS}=t=t_1\cdots t_N(t_{N+1}\cdots t_{N+P})^{\infty}.$$

证明完毕．

现在举一个例子，用来说明上面证明过程中的主要步骤．

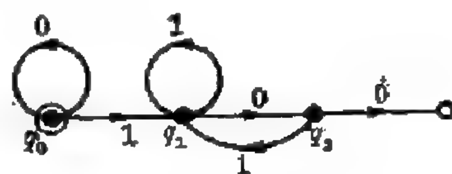


图 8-1

［例 4］ 已知图 8-1 中的有限自动机所接受的语言是由某个揉序列 KS 所确定的，求 KS．这时 $F=\{q_0,\ q_1,\ q_2\}$．按照定理中的方法，可以求出极大路径所对应的序列

$$t=101101\cdots,$$

同时写出关于结点和弧的相应序列：

$$q_0\xrightarrow{1}q_1\xrightarrow{0}q_2\xrightarrow{1}q_1\xrightarrow{1}q_1\xrightarrow{0}q_2\xrightarrow{1}\cdots.$$

其中第二个状态 q_1 和第五个状态 q_1 符合在定理证明中的第一种情况的条件，这样就得到

$$t=1(011)^{\infty}.$$

实际上容易看出 t 是周期序列$(101)^{\infty}$．这就是所要求的揉序列．

关于定理 1 已知有两个证明方法．这里介绍的是在[35]、[36]中的证法．在[38]中完全从 R_L 等价类的分析出发给出了另一个证明．它不像上述证明那样直观，但在方法上更为一般，可用于解决一些其他问题，这里从略．

§8.4 逆定理及其意义

定理 1 的逆定理也成立．由于关于周期揉序列和终极周期揉

序列的情况全然不同，我们将逆定理分成两个定理来讨论.

定理2 如果揉序列 KS 为周期序列，则 $\mathscr{L}$(KS)为正规语言.

定理3 如果揉序列 KS 为终极周期序列，但不是周期序列，则 $\mathscr{L}$(KS)为正规语言.

下面讨论这两个定理的意义.

举 KS$=(101)^{\infty}$ 为例. 接受语言 $\mathscr{L}((101)^{\infty})$的有限自动机已在图 8-1 中给出. 从 §9 的讨论我们会知道，这即是接受 $\mathscr{L}((101)^{\infty})$的最小确定性有限自动机.

设 f 是以$(101)^{\infty}$为揉序列的一个单峰映射. 引用 §6.3 的定理 2，我们知道存在以 $(101)^{\infty}$ 为踪迹的周期 3 轨. 根据沙可夫斯基定理或李-约克定理(见[19]、[18])，单峰映射 f 具有任何周期的周期轨. 这个事实也可以用下面的方法直接得到：先验证 0^{∞}、1^{∞}、$(10)^{\infty}$ 和$(101^{n-2})^{\infty}$($n\geqslant 3$)都是满足 §6.1 条件的允许字，然后再用 §6.3 的定理 2，知道存在相应的所有周期轨. 这些允许字也很容易从图 8-1 的有限自动机中直接找到. 实际上，还存在许多其他的周期轨.

从[18]我们知道，除了上述周期轨之外，还出现复杂的混沌行为，即所谓在李-约克意义下的混沌(参见[22]、[24]等). 这类动力学行为在图 8-1 的自动机中也有反映. 实际上，这里的自动机的主要部分与[18]中的几何证明方法所用的图是非常相似的，虽然它们的具体意义并不一样. 定理 2 表明，从形式语言的角度看，这时的复杂性仍是属于简单的层次，即正规语言.

对于二次方映射$f(x)=bx(1-x)$，或者满足施瓦兹导数为负条件的其他单峰映射，当揉序列为周期序列时，存在唯一的稳定周期轨. 同时，除去一个勒贝格测度为零的集合之外，点 x 出发的轨 $f^{n}(x)$都收敛于这个稳定的周期轨. 这些结论当然不能用形式语言工具得到. 我们在这里研究的实际上是以某个给定的移位最大字为揉序列的所有单峰映射的共同性质. 例如，在 KS$=(101)^{\infty}$

时的单峰映射可以有多于一个的稳定周期轨，甚至有无穷多个稳定周期轨. 这样的例子不难找到. 但就符号动力学行为而言，则都是相同的.

定理3所讨论的情况与定理2完全不同，即所谓粗粒混沌的情况(参见[26]). 其中最简单的例子即是 $KS=10^{\infty}$，已在§7.1中作过研究. 讨论得最为透彻的具体映射是二次方映射族中的满射情况，即

$$f(x)=4x(1-x), \qquad 0\leqslant x\leqslant 1$$

和帐蓬映射族中的满射情况，即

$$F(x)=\begin{cases}2x, & \text{当 } 0\leqslant x\leqslant \dfrac{1}{2} \text{ 时;}\\ 2(1-x), & \text{当 } \dfrac{1}{2}<x\leqslant 1 \text{ 时.}\end{cases}$$

在图8-2中作出了 $F(x)$ 的图象.

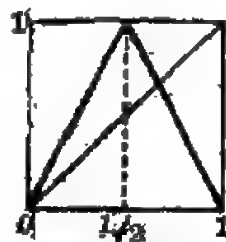


图 8-2

粗略地说，这时的轨道行为具有极大的多样性，符合关于混沌的各种定义. 这方面可以参见[24]和讨论吸引子、混沌概念的论文[50]、[51]等. 定理3表明，从语言复杂性的角度来看，系统的符号动力学行为从整体上看仍属于简单层次，即正规语言.

在[37]中证明了，定理3中的正规语言，除了 $\mathscr{L}(10^{\infty})=S^{*}$ 这个唯一例外，在某种确定的意义上要比定理2中的正规语言复杂. 这个问题将在本书第5章中讨论，这里的差别在于，在定理3中的正规语言，除 $\mathscr{L}(10^{\infty})$ 之外，属于所谓无限补语言(Infinite Complement Languages)，而在定理2中的正规语言则属于有限补语言(Finite Complement Languages).

§8.5 文献简述

关于定理2和定理3，在文献中已有许多讨论. 在[40]中，格拉斯贝尔格(P. Grassberger)利用[52]中的有向图方法指出，当

KS 为周期序列时，相应的语言为有限补语言，从而是正规语言．他在[41]中给出了构造有限自动机的方法，举出了 KS 为周期序列和终极周期序列时的最小自动机的大量例子．其中包括揉序列为

$$101^{\infty}, \quad 100(101)^{\infty}, \quad 1011(10)^{\infty}$$

的情况，分别对应于混沌带从二到一，从三到一和从四到二的汇合点[†]．在[41]还给出了 $KS=101^{\infty}$ 和 $1011(10)^{\infty}$ 两种情况的马尔可夫划分．

类似的结果在[43]、[44]中也可找到．在[45]中讨论了 $KS=(10010110)^{\infty}$ 的有向图，作出了 $KS=100(101)^{\infty}$ 的有限自动机．在[46]中作出了 $KS=101(10)^{\infty}$ 的有限自动机．在[47]中证明了 KS 为周期序列时 $\mathscr{L}$(KS)为正规语言，作出了 $KS=(10c)^{\infty}$ 的最小有限自动机．在[49]中对 $KS=101^{\infty}$ 作出了马尔可夫划分，并讨论了在区间映射中寻找非正规语言的可能途径，这对本书下一章的讨论起了重要作用．

本书作者和同事们在[35]、[36]中证明了本节的三个定理，在[37]研究了两类正规语言之间的区别(见 § 8.4 末)，在[38]中采用 R_L 等价类方法对三个定理作出新的证明，同时对于 KS 为周期序列时的最小有限自动机问题作出了理论上的完整处理，这方面的结论在[42]中已经猜测到了．关于 KS 为终极周期序列时的最小有限自动机问题已于本书写作期间得到解决([39])．

§ 8.6 马尔可夫划分方法

这一小节介绍在[45]中关于 $KS=100(101)^{\infty}$ 的分析，其中所用的马尔可夫划分方法具有一定的代表性．

由于二次方映射 $f(x)=bx(1-x)$ 中当参数 b 从 0 变化到 4 时可以实现任何单峰映射的揉序列(参见[22])，因此不妨限于对

† 在[41]的第 675 页上图 3 中关于 $KS=100(101)^{\infty}$ 的最小有限自动机有误，实际上它的状态个数还可以减少．参见本书下面的图 8-5 和 § 9．

某个二次方映射来研究 KS = 100(101)$^\infty$. 实际上相应的参数值为 3.857…. 这时不存在稳定周期轨，从临界点出发的轨只能是直接落到踪迹为 (101)$^\infty$ 的一个不稳定周期轨上. 这样一来，轨 $f^n(c)$ 只含有限个不同的点，它们将区间 [0, 1] 划分成有限个子区间，如图 8-3 所示. 其中点 1～6 分别表示点 $f(c)$、…、$f^6(c)$，$f^4(c)$、$f^5(c)$ 与 $f^6(c)$ 为周期 3 点. 现在用字母 A～E 表示各个子区间:

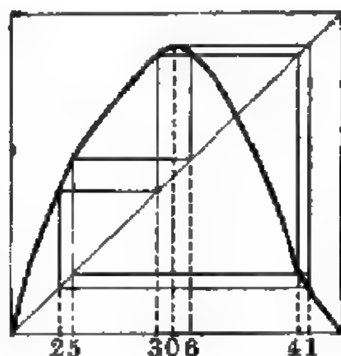


图 8-3

$A=[f^2(c), f^5(c)]$, $B=[f^5(c), f^3(c)]$, $C=[f^3(c), c]$, $D=[c, f^6(c)]$, $E=[f^6(c), f^4(c)]$, $F=[f^4(c), f(c)]$. 这些区间中的每一个区间在 f 映射下的象是这些区间中的一个或几个的并. 我们称这样的划分为马尔可夫划分.

利用马尔可夫划分可以求出接受语言 $\mathscr{L}$(KS) 的有限自动机. 按照[45]的方法，考虑每一个区间在映射 f 下的原象. 以 0 代表 f 的左枝，以 1 代表 f 的右枝，写出

$$A \xrightarrow{0} C、D, \qquad B \xrightarrow{0} E,$$

$$C \xrightarrow{0} F, \qquad D \xrightarrow{0} F,$$

$$E \xrightarrow{1} B、C、D、E, \qquad F \xrightarrow{1} A.$$

以其中第一个为例说明它们的意义: $A \xrightarrow{0} C、D$ 表示在 f 左枝的映射下区间 A 的象是区间 C 和 D 的并. 利用以上信息，作出图 8-4 的方向图. 其中箭头方向与上面的表达式相反，因此沿着与

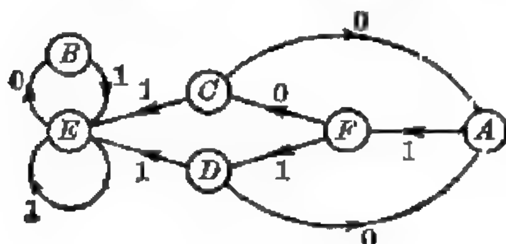


图 8-4

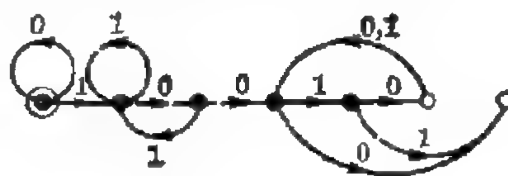


图 8-5

箭头相反方向的路径代表动力系统的实际演化. 在[45]用这个图来研究周期轨,因此方向相反也没有关系.

如果将这个有向图中的箭头方向取反,再利用[1]中的方法求出最小确定性有限自动机,就得到图 8-5 的结果. 这与[37]中的方法所得的结果完全一样. 这个方法将在第 9 节中介绍.

§8.7 关于揉序列前缀的研究

这里介绍[38]中的基本结果. 它给出了揉序列与 R_{L} 指标之间的直接关系. 利用这个结果就很容易证明定理 2 和 3. 并且可以对定理 1 作出新的证明.

定理 4 $\mathrm{L}=\mathscr{L}(\mathrm{KS})$为正规语言的充分必要条件是在 KS 的前缀中至少有两个是 R_{L} 等价的.

证明: 必要性是容易的. 如 $\mathrm{L}=\mathscr{L}(\mathrm{KS})$为正规语言,则由迈希尔-奈罗德定理,$R_{\mathrm{L}}$ 等价类只有有限个,因此结论显然成立. 关于充分性,首先证明,如果 x、$y\in\mathrm{L}$ 是两个串,它们关于 KS 有相同的最长前后缀,那么成立

$$xR_{\mathrm{L}}y.$$

实际上,记它们关于 KS 的最长前后缀为 v, 写出 $x=uv$, $y=u'v$, 然后就可以对于任何串 $z\in S^*$,建立

$$xz\in\mathrm{L}\Longleftrightarrow yz\in\mathrm{L}.$$

这里的具体细节与§7.5 和§7.7 相同,从略.

然后, 在 R_{L} 等价类中除了[ε]与 L′ 之外(参见§8.2), 我们可以证明,每一个等价类中至少含有 KS 的一个非空前缀.

对于 $\mathrm{KS}=10^{\infty}$ 和 0^{∞} 的情况, 在[ε]和 L′ 之外没有 R_{L} 等价类,因此不必讨论. 对于其他揉序列的情况,由于[ε]中已吸收了所有 0 串,因此,如果记[x]为在[ε]和 L′ 之外的一个 R_{L} 等价类,则代表元 x 必含符号 1. 利用§7.6 中关于前后缀存在问题的讨论, x 一定有关于 KS 的最长前后缀,记为 v,则由上面的讨论,就

得到 $xR_{L}v$ 由于 v 为 KS 的一个前缀，因此已得到所要证的结论：$v\in[x]$.

现在设 $KS=x_1x_2\cdots x_n\cdots$，并将第 n 个前缀记为

$$x^{(n)}=x_1x_2\cdots x_n,\qquad n\geqslant 1.$$

假设定理中的条件成立，即存在 KS 的两个不同的前缀 $x^{(i)}$ 和 $x^{(j)}$，$1\leqslant i<j$，使 $x^{(i)}R_Lx^{(j)}$ 成立．我们可以证明，对于 $l\geqslant 1$ 的每个前缀 $x^{(j+l)}$，都与 $x^{(1)}$、$x^{(2)}$、$\cdots$、$x^{(j-1)}$ 中的某一个是 R_L 等价的．为此，对 l 用数学归纳法．

先讨论 $l=1$．如果 $x_{i+1}=x_{j+1}$，则从 R_L 的右不变性质(见§1.8)，有

$$x^{(i)}R_Lx^{(j)}\Rightarrow x^{(i+1)}R_Lx^{(j+1)}.$$

如果 $i+1=j$，则 $x^{(j+1)}$ 与 $x^{(i)}$ 为 R_L 等价．如 $i+1<j$，则已无问题．

如果 $\bar{x}_{i+1}=x_{j+1}$，则有

$$x^{(j+1)}R_Lx^{(i)}\bar{x}_{i+1}.$$

由于 $x^{(i)}\bar{x}_{i+1}$ 不是 KS 的前缀，利用§7.6 取出串 $x^{(i)}\bar{x}_{i+1}$ 关于 KS 的最长前后缀，记为 $x^{(i')}$，则有 $i'\leqslant i$．这时有 $x^{(j+1)}R_Lx^{(i')}$．$l=1$ 已讨论完毕．

从上述论断对 l 成立而推出对 $l+1$ 也成立的方法与 $l=1$ 的讨论完全一样，这里从略．

这就可以知道，採序列 KS 的无穷多个前缀最多分属于 $j-1$ 个 R_L 等价类：

$$[x^{(1)}],\quad [x^{(2)}],\quad \cdots,\quad [x^{(j-1)}].$$

当然，这里不排除其中有相等的可能．

由前面确立的事实，我们知道除$[\varepsilon]$和 L' 之外已没有不包含 KS 的前缀的等价类，因此 R_L 的指标 N 满足不等式：

$$N\leqslant 2+(j-1)=j+1.$$

从 N 有限即推出 $\mathscr{L}(KS)$为正规语言．证毕．

利用定理 4 可以证明定理 2 和 3 而不必利用马尔可夫划分，

具体的证明细节写在附录 B 中. 在定理 4 证明中得到的不等式 $N \leqslant j+1$ 也是有用的结果.

§9 最小有限自动机的构造

本节取材于[38]、[39],主要说明方法,并举大量例子.由于所涉及到的证明过程甚长,只能介绍有关结果,细节从略.所需的知识见§1.

§9.1 构造自动机的基本方法

今以 $\mathrm{KS}=x^{\infty}$ 为例,来说明这个方法.

在第 1 章介绍的各类自动机中,可以将有限自动机看成是不需要存储器的最简单的一类自动机.但实际上在用有限自动机识别符号串时并非不要记忆,只不过这里的记忆容量为有限,可以用控制器的状态来实现记忆,而无需其他额外的存储装置.

利用附录 B 中 B.1 的引理可知,当 $\mathrm{KS}=x^{\infty}$,而且 x 为偶串时,自动机需要记住的符号串的长度不超过串 x 的长度.

举个具体例子.讨论 $\mathscr{L}((101^{\infty}))$.设有一个有限自动机 M 能识别 $\mathscr{L}((101)^{\infty})$.现在用 M 来识别 S^{*} 中的一个符号串

$$z=z_1z_2\cdots z_n\cdots,$$

如果 $z_1=0$,则不必记住,而当 $z_1=1$ 时则需要记住它.如果 $z_1=z_2=1$,则只要记住 $z_2=1$ 就够了.但如果是 $z_1=1$, $z_2=0$,则要将 $z_1z_2=10$ 一起记住.以上这些考虑的理由很简单,因为串 $100\notin\mathscr{L}((101)^{\infty})$,如果自动机 M 不记住已出现的 10 串,就不可能对 $z_3=0$ 时作出不接受的反应,同时又对 $z_3=1$ 作出接受的反应.同样可看出,对于 $z_1z_2z_3z_4=1010$ 的情况,只要记住 $z_3z_4=10$ 也就够了.这样就可以得出图 9-1(i)中的自动机 M.在[35]、[36]中应用相同的方法,对于 $\mathrm{KS}=x^{\infty}$ 的一般情况构造了接受 $\mathscr{L}(x^{\infty})$ 的有限自动机,其中采用了§1.6 中介绍的右线性语法.

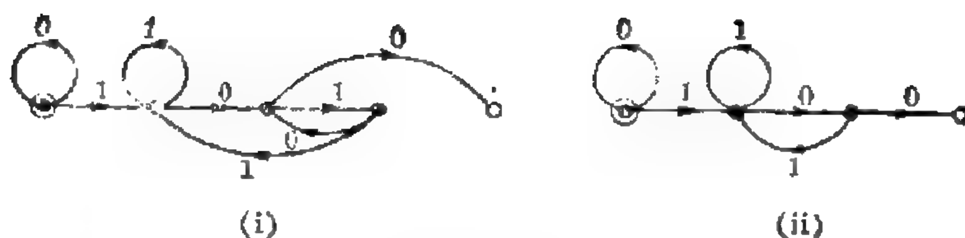


图 9-1

对(i)可写出相应的右线性语法. 令

$$G=(V,\ S,\ P,\ s_0),$$

其中

$$V=\{s_0,\ s_1,\ s_2,\ s_3\}$$

分别对应于图(i)中的四个终止状态, $S=\{0,\ 1\}$, 语法规则

$$P=\{s_0\to\varepsilon\,|\,0s_0\,|\,1s_1,\ s_1\to\varepsilon\,|\,1s_1\,|\,0s_2,$$
$$s_2\to\varepsilon\,|\,1s_3,\ s_3\to\varepsilon\,|\,0s_2\,|\,1s_1\}.$$

不难看出, 使用右线性语法和状态转移图在这里是一回事.

但以上得到的(i)还不是最小有限自动机. 实际上, 如果注意到在输入 $z_1z_2z_3=101$ 时只要记住 $z_3=1$, 就可以得到(ii)中所示的最小有限自动机.

一般来说, 可以用[1]中的算法, 从接受 L 的某一个有限自动机出发, 求出接受 L 的最小确定性有限自动机. 在这里介绍一个简单命题, 利用它就可以在很多情况中从某一个状态转移图出发, 进行某种图上作业而得到最小自动机. 这里又利用了 R_L 等价类与状态的等同性.

命题 设 x、$y\in S^*$. 如对每个 $a\in S$ 有 xaR_Lya 成立, 那么 xR_Ly.

证明 对任意串 $z\in S^*$ 考虑 xz 和 yz. 如果 $z\neq\varepsilon$, 则可写 $z=az'$, $a\in S$. 从条件 xaR_Lya 和 R_L 的右不变性即推出 xzR_Lyz, 因此串 xz 和 yz 必是同属于 L 或都不属于 L.

现观察 $z=\varepsilon$ 的情况. 如果 x 与 y 中有一个属于 L, 另一个不属于 L. 例如, 设 $x\in$ L, $y\notin$ L. 利用 §7.4 性质 2, 存在 $a\in S$, 使

$xa \in L$. 但从 § 7.4 的性质 1, 则有 $y \notin L \Rightarrow ya \notin L$. 这与 $xaR_L ya$ 矛盾.

根据对 xz 和 yz 的讨论, 可见 $xR_L y$. 证毕.

现在利用在有限自动机讨论中的一个基本方法, 即将每一个状态(或结点)与输入符号串的一个集合等同起来, 这个集合即是从初态到该状态的所有可能输入串所构成的, 它在 R_L 的一个等价类中. 这样就可以在状态转移图上应用上面这个看起来似乎有些抽象的命题.

举例来说, 在图 9-1(i)中, 与 s_1 和 s_3 对应的两个结点, 沿着标以 0 的弧都到达 s_2, 而沿着标以 1 的弧则都到达 s_1. 应用上述命题, 合并 s_1 和 s_3, 这就得到图 9-1(ii).

注 有时仅仅使用上述命题还不足以使有限自动机(的状态个数)达到最小, 而还需要其他命题(见 § 19.2). 另一个主要问题是, 如何判定一个有限自动机已是最小. 这方面可参看[1]. 对于本章的两类正规语言, 则已建立了一般性的理论结果, 即以下的主要内容.

§ 9.2 周期情况的最小自动机

在正规语言类中比较复杂程度时, 可以将接受正规语言的最小有限自动机的状态个数作为复杂性的一个度量. 这是沃尔夫勒姆(S. Wolfram)在研究元胞自动机的论文[53]中首先提出来的. 实际上, 这也是将计算机科学中的形式语言等工具用于动力系统研究的最早文献.

设 $KS = x^\infty$, 研究接受 $\mathscr{L}(x^\infty)$的最小有限自动机的构造方法.

由于 x^∞ 也可以写成$(x^2)^\infty$ 或$(x^3)^\infty$ 等等, 因此如果要用串 x 的长度将最小自动机的状态个数表示出来, 则对 x 本身必定要加一定的限制.

这里提出极小串的概念. 即要求 x 为偶串, 同时串 x 的长度达到极小. 利用附录 B.3 中的既约串概念, 可知极小串只有两种:

1. x 为偶既约串.

2. $x=y^2$, y 为奇既约串.

这里的既约串即是不能写成较短串的重复连接的意思.

引用[38]中的结果, 即有

定理 1 如果 $\mathrm{KS}=x^\infty$, x 为极小串, 那么接受语言 $\mathscr{L}(x^\infty)$ 的最小有限自动机的状态个数为

$$N=|x|+1.$$

简要说明一下证明的主要步骤和定理 1 的应用.

设 $|x|=n$, 记

$$x=x_1x_2\cdots x_n,$$

记 x 的前缀为

$$x^{(i)}=x_1x_2\cdots x_i,\qquad 1\leqslant i\leqslant n.$$

主要困难在于证明, $x^{(1)}$、$x^{(2)}$、$\cdots$、$x^{(n-1)}$ 这 $n-1$ 个前缀中任何两个都不是 R_L 等价的, 同时, $x^{(n)}(=x)$ 则与它们中的某一个等价. 然后引用 § 8 的定理 4, 就得到 $N=(n-1)+2=n+1$.

为了构造最小有限自动机, 需要知道 $x^{(n)}(=x)$ 和长度比它短的哪一个前缀是 R_L 等价的. 它的计算方法如下: 不妨设 x 不是 0 串, 因此从 x 为偶串知它至少含两个符号 1. 去掉 x 的第一个符号 1, 求 $x_2\cdots x_n$ 关于 KS 的最长前后缀, 记为 $x^{(n')}$, 则 $1\leqslant n'\leqslant n-1$. 于[38]中已证明 $xR_Lx^{(n')}$.

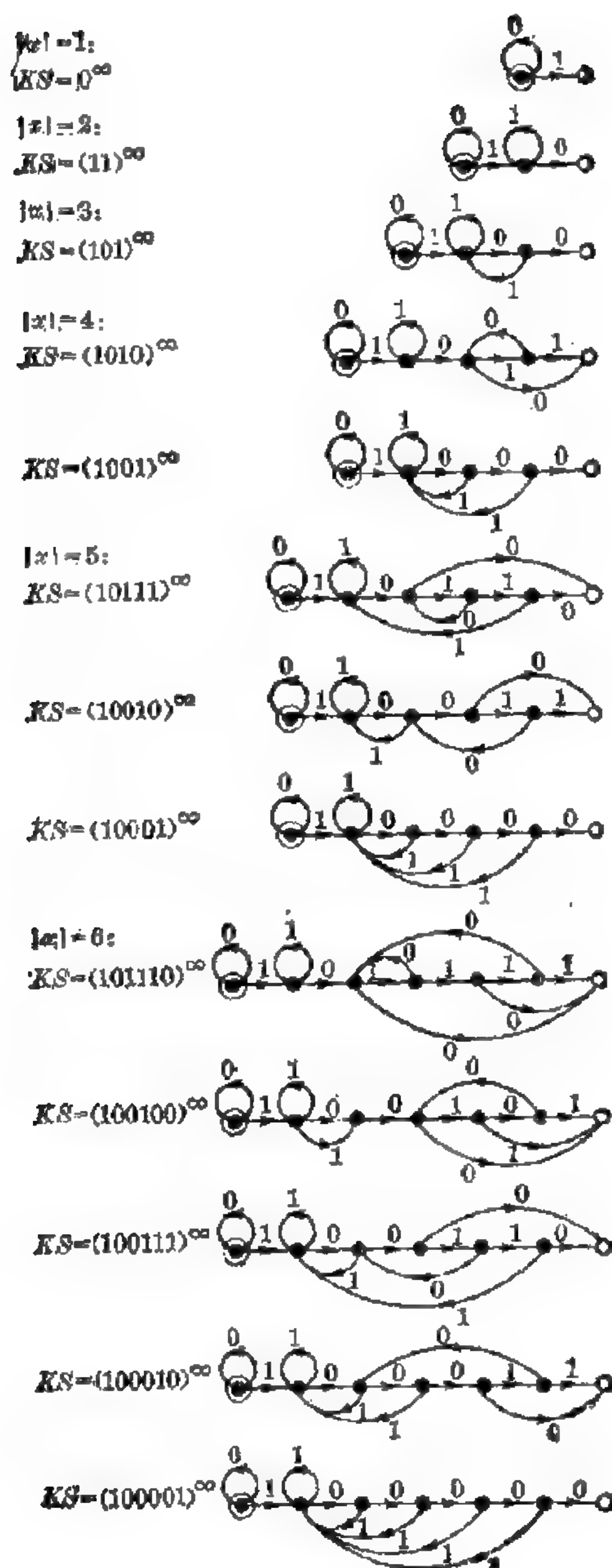
现在可以直接写出接受 $\mathscr{L}(x^\infty)$ 的最小有限自动机

$$M=(Q,\ S,\ \delta,\ q_0,\ F),$$

其中 $Q=\{q_0,\ q_1,\ \cdots,\ q_{n-1},\ q_n\}$, q_0 为初态, $F=\{q_0,\ q_1,\cdots,q_{n-1}\}$, q_n 为唯一的非终止态, $S=\{0,\ 1\}$, δ 是从 $Q\times S$ 映入 Q 的转移函数, 定义如下:

1. $\delta(q_i,\ x_{i+1})=q_{i+1}$, $0\leqslant i\leqslant n-2$.

2. $\delta(q_0,\ 0)=q_0$. 对于 $1\leqslant i\leqslant n-2$, 如果符号串 $x^{(i)}\bar{x}_{i+1}\in \mathrm{L}$, 则定义 $\delta(q_i,\ \bar{x}_{i+1})=q_{f(i)}$, 其中 $f(i)$ 是 $x^{(i)}\bar{x}_{i+1}$ 关于 KS 的最长前后缀的长度. 如果 $x^{(i)}\bar{x}_{i+1}\notin \mathrm{L}$, 则定义 $\delta(q_i,\ \bar{x}_{i+1})=q_n$.



|x|=1:
KS=0^∞
|x|=2:
KS=(11)^∞
|x|=3:
KS=(101)^∞
|x|=4:
KS=(1010)^∞
KS=(1001)^∞
|x|=5:
KS=(10111)^∞
KS=(10010)^∞
KS=(10001)^∞
|x|=6:
KS=(101110)^∞
KS=(100100)^∞
KS=(100111)^∞
KS=(100010)^∞
KS=(100001)^∞

图 9-2

3. $\delta(q_{n-1}, x_n)=q_{n'}$, n' 即串 $x_2\cdots x_n$ 关于 KS 的最长前后缀的长度.

4. $\delta(q_{n-1}, \bar{x}_n)=q_n$.

5. $\delta(q_n, 0)=q_n$, $\delta(q_n, 1)=q_n$.

当然,也容易写出生成 $\mathscr{L}(x^\infty)$ 的右线性语法,这里从略.

§9.3 例子

对于 $KS=x^\infty$, x 为极小串的情况. 在图 9-2 中作出了 $|x|$ 从 1 到 6 的所有最小有限自动机. 其中虽有少数曾于以前各节中已给出过,但为了便于参考比较,仍列出在这里.

对图 9-2 作些说明. 关于非终止状态,从 §8.1 中的讨论可见不必作出由它出发的弧. 此外,在图 9-2 中的 $KS=x^\infty$ 均按 x 为极小串的要求写出,因此将 1^∞ 写为 (11^∞), 将 $(10)^\infty$ 写成 $(1010)^\infty$, 将 $(100)^\infty$ 写成 $(100100)^\infty$, 只有这样才能从串 x 的长度反映出接受语言 $\mathscr{L}(x^\infty)$ 的最小有限自动机的状态的个数.

对于状态个数相同的最小有限自动机, 我们可以利用第 5 章中的方法进一步比较它们之间的复杂程度.

还可以看出,当 $KS=x^\infty$ 中的 x 为第二类极小串, 即 $x=y^2$, y 为既约奇串时,相应的最小自动机所具有的一些特点.

这时容易证明在上一小节中的 $n'=n/2=|y|$, 也就是 yR_Ly^2. 又可以证明, 对 $n'\leqslant i\leqslant n-1$ 的每个 i, 有 $x^{(i)}\bar{x}_{i+1}\notin L$, 即 $\delta(q_i, \bar{x}_{i+1})=q_n$. 从而在状态转移图中从 $q_{n'}$ 开始的 $q_{n'}$、$q_{n'+1}$、$\cdots$、q_{n-1} 再回到 $q_{n'}$ 构成一个闭回路, 其中每一个状态只有一条弧通向终止状态. 这表明当在 $KS=y^\infty$ 的允许字中, 如出现子串 y, 则以后只能是 y 的重复.

§9.4 终极周期情况的最小自动机

关于 $KS=10^\infty$ 和 $100(101)^\infty$ 的情况, 已在 §7.1 和 §8.6 作过讨论. 这里处理一般情况.

由于一个终极周期揉序列在写成 $\rho\lambda^\infty$ 的形式时，ρ 和 λ 的取法有很大的随意性，因此如何确定 ρ 与 λ，一方面使 $KS=\rho\lambda^\infty$，另一方面又与接受 $\mathscr{L}(KS)$ 的最小有限自动机有直接联系，这就成为这里的主要问题．以下材料取自[39]．

如果 KS 为终极周期序列，但不是周期序列，则已经证明，一定存在一对符号串 ρ 和 λ，使 $KS=\rho\lambda^\infty$，同时满足下面三个条件：

1° ρ 为奇串，λ 为偶串．

2° ρ 和 λ 都没有关于 KS 的偶前后缀．

3° 在满足条件 1° 和 2° 的前提下，ρ 和 λ 的长度分别达到极小．

如果 $KS=\rho\lambda^\infty$，其中的 ρ 和 λ 满足条件 1° 和 2°，则称此表达式为恰当形式．如果条件 3° 也满足，则称为最小恰当形式．

定理2 设揉序列 $KS=\rho\lambda^\infty$ 为最小恰当形式，那么接受 $\mathscr{L}(KS)$ 的最小有限自动机的状态个数为

$$N=|\rho\lambda|+1.$$

由于这个定理的证明涉及到很多技术细节，这里从略，下面只说明如何从一个给定的 KS 出发求出其最小恰当形式，然后又如何构造出相应的最小有限自动机，并用一个例子来说明．

首先，将 KS 写成 ρ 和 λ 都为最短的形式．这时 ρ 和 λ 的最后一个符号一定不相同，否则还可以进一步降低 ρ 的长度．λ 的最短长度是由终极周期序列的周期部分的最小周期唯一确定的．

［例 1］ 如一开始有

$$KS=1011(01)^\infty,$$

其中 $\rho=1011$，$\lambda=01$，最后一个符号均为 1，则可改写为

$$KS=101(10)^\infty.$$

由于 KS 不是周期序列，有限次改写后一定可以达到要求．

其次，在 λ 为奇串时用 λ^2 代替 λ．对上述例子，即有

$$KS=101(1010)^\infty.$$

然后，检验条件 1° 中 ρ 为奇串以及条件 2° 是否满足．如不

满足，则将 ρ 的长度加 1，相应地改写 λ，再作检验，直到满足为止。例如上面的形式中 $\rho=101$ 不是奇串，则改写为

$$\mathrm{KS}=1011(0101)^{\infty}.$$

这时 $\lambda=0101$ 有关于 KS 的偶前后缀 101，因此不满足条件 2°。再改写为

$$\mathrm{KS}=10110(1010)^{\infty}.$$

这时已为最小恰当形式.

由[39]中的理论性讨论，以上方法在有限次后一定可以达到目的.

现说明如何利用这个形式直接写出接受 $\mathscr{L}(\mathrm{KS})$ 的最小有限自动机.

设 $\mathrm{KS}=\rho\lambda^{\infty}$ 为最小恰当形式. 在[39]中已经证明, KS 的长度小于 $|\rho\lambda|$ 的所有前缀中的任何两个都不是 R_L 等价的，同时已证明 $\rho\lambda R_L\rho$ 一定成立. 采用在 § 9.2 中的相同方法，就可以作出接受 $\mathscr{L}(\mathrm{KS})$ 的一个有限自动机. 这里知道 $\rho\lambda R_L\rho$ 是关键的一步. 在[39]中的理论研究保证了所得到的即是最小有限自动机.

对于例 1，已有

$$\mathrm{KS}=10110(1010)^{\infty},$$

可知道状态个数 $N=10$. 仿照 § 9.2 的方法，利用 $101101010R_L10110$，就可作出图 9-3 中的最小自动机.

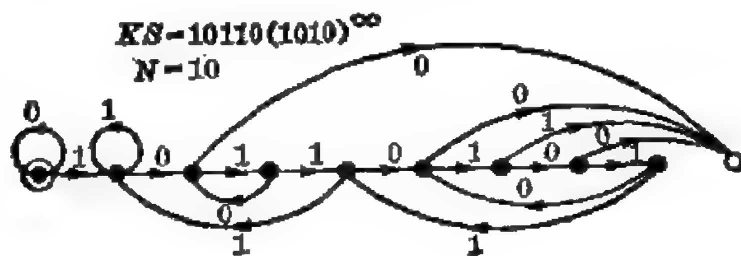


图 9-3

关于构造有限自动机的方法可参看 § 8.7 中的定理 4.

再举几个例子.

[例 2]　$KS=101^{\infty}$.

令 $\rho=10$, $\lambda=11$, 则 $KS=10(11)^{\infty}$ 已是最小恰当形式. 这时 $N=5$. 除了 $KS=10^{\infty}$ 之外，这是在终极周期揉序列中最简单的情况. 这里所谓简单即是从最小有限自动机的状态个数最少来说的. 它的最小自动机如下图所示.

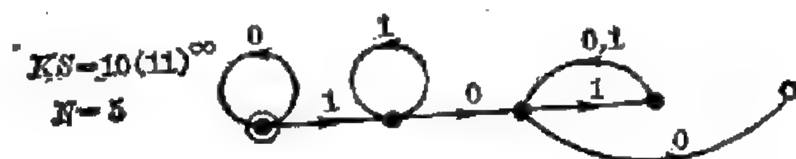


图 9-4

我们知道，$KS=101^{\infty}$ 对应于二次方映射族的两片混沌带汇合成一片时的情况. 这时的系统除不动点外只有偶周期轨. 这在图 9-4 上也是很明显的. 以 101^{∞} 为揉序列的单峰映射可以具有各种不同特征，但只具有偶周期轨则是它们共有的性质. 利用图 9-4 和 § 6.3 的定理，我们从揉序列出发直接导出了这个一般性结论.

[例 3]　$KS=100(101)^{\infty}$. 这里的 KS 已为最小恰当形式, $N=7$, 见图 8-5. 它对应了混沌带从三片汇合成一片的激变情况.

[例 4]　$KS=10(0101)^{\infty}$. 如写成 $KS=100(1010)^{\infty}$, 则为恰当形式而不是最小恰当形式. 它的最小自动机如下图所示, $N=7$.

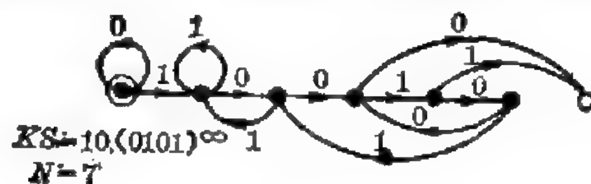


图 9-5

§ 9.5　*合成律与广义合成律

对二次方映射来说，每一个周期窗口左方的揉序列为 $KS=x^{\infty}$, 其中 x 为偶既约串. 由此可以生成 * 合成律，它可以表示为符号变换规则

$$h=\{0\to x,\ 1\to \hat{x}\},$$

其中 $\hat{x}$ 是将串 x 的最后一个符号取反码后得到的奇串. 利用 h 容易揭示出分岔图在每一个周期窗口中的自相似性质(参见[26]).

在[26]中考虑了更一般的广义合成律.

$$W=\{0\to\lambda,\ 1\to\rho\},$$

它包含 * 合成律为特例, 且有多方面的应用. 这里的符号串 ρ 和 λ 要满足以下三个条件:

i. ρ 为奇串, λ 为偶串;

ii. $\rho>\lambda$;

iii. $\rho|_c$、$\rho\lambda|_c$ 和 $\rho\lambda^\infty$ 均为揉序列.

这里记号 $\rho|_c$ 和 $\rho\lambda|_c$ 是将最后一个符号改为 c 所得到的符号串, 同时这里对含有符号 c 的揉序列采取只写到 c 为止的有限记法.

在[26]中称 $\rho\lambda^\infty$ 为粗粒混沌, 它具有与 $KS=10^\infty$ 相似的一系列性质.

利用上一小节中的结果, 容易证明

定理 8 如果 KS 是终极周期揉序列, 但不是周期序列, 那么存在满足广义合成律要求的 ρ 和 λ, 使 $KS=\rho\lambda^\infty$.

这里解释一下证明方法及其意义.

实际上, 利用在[39]中关于恰当形式的存在性, 容易证明当 ρ 和 λ 为恰当形式时, 它们就满足广义合成律中的所有条件.

由此出发, 利用广义合成律, 即可得到

$$\mathscr{L}(\rho\lambda^\infty)\sqsupseteq(\rho+\lambda)^*.$$

但这里的 ρ 和 λ 必须满足恰当形式中的条件. 例如在§9.4中的例 1, 就有

$$\mathscr{L}(10110(1010)^\infty)\sqsupseteq(10110+1010)^*.$$

但如将同一个 KS 写为 $101(10)^\infty$, 则不能得到

$$\mathscr{L}(101(10)^\infty)\sqsupseteq(101+10)^*$$

的结论. 实际上, 例如 10110110 就不是语言中的字, 因此这个结

论是错误的.

这样我们就可以说，每一个终极周期揉序列(但不是周期序列)，就代表了一个粗粒混沌. 它同样导致在分岔图内部的一个自相似结构，只是在几何直观上不如周期窗口那样明显.

如果将周期运动称为规则运动，而将粗粒混沌看成为另一个极端，那么利用形式语言的工具已知道，处于这两个极端的系统的复杂性层次都是最简单的正规语言. 在第8节中的三个定理以及这一小节中的内容对于这个结论作出了严格的数学讨论，同时对于处于两端的系统行为给以确切的数学描述. 这里的规则运动是指揉序列为周期序列的系统. 如果它又满足施瓦兹导数为负的条件，那么从几乎处处的点出发的轨都收敛于唯一的稳定周期轨. 对于粗粒混沌情况，如果满足一定的条件，则存在绝对连续的遍历不变测度，它反映出从几乎处处的点出发的轨都具有极强的随机性质. 在图9-6中将上述讨论用一条曲线表示出来. C 是对复杂性的某种度量，H 是对随机性的某种归一化度量. 当 H 很小或很大时 C 很低. 真正复杂的系统介于两个极端之间(参看[44]).

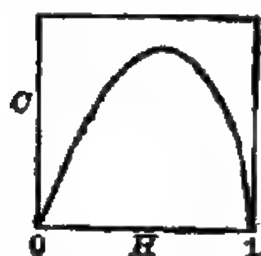


图 9-6

第4章

区间映射中的非正规语言

在这方面的研究还只是刚才开始，像第8、9节关于正规语言那样的理论还有待建立. 因此在这一章中主要是分析已经研究过的一些重要例子，提出一些有关的问题.

在第10、11节详细分析了费根鲍姆吸引子(Feigenbaum Attractor)的语言及其复杂性。在第12节将这个结果推广到更为一般的倍周期分岔和n周期分岔的极限情况. 然后介绍在所谓斐波那契序列方面的研究. 在最后部分还讨论了出现各种复杂性层次的可能性及有关问题.

§10 费根鲍姆吸引子的形式语言

本节介绍区间映射中的一个重要情况，即倍周期分岔的极限情况，一般称为费根鲍姆吸引子. 利用符号动力学方法给出这时的揉序列t_∞，对语言$\mathscr{L}(t_\infty)$的结构作出分析，为在下一节对$\mathscr{L}(t_\infty)$的复杂性分析作好准备. 本节材料主要来自[54]，进一步的推广见[55].

§10.1 倍周期分岔的极限

为了理解什么是费根鲍姆吸引子，我们观察图10-1. 它是二次方映射$f(x)=bx(1-x)$的参数分岔图，其中$0\leqslant x\leqslant 1, 0\leqslant b\leqslant 4$. 利用微型计算机很容易制成这样的分岔图. 例如，对参数b从1开始每增长0.01计算一次，直到$b=4$. 对上述每个b值，从初值$x_0=0.5$开始按迭代公式$x_n=bx_{n-1}(1-x_{n-1})$，$n\geqslant 1$，计算轨

$$f^{n}(x_0)=(x_0,\ x_1,\ \cdots,\ x_n,\ \cdots).$$

不妨设每次计算到 $n=1000$ 为止. 将其中前 100 点弃去, 而将其余点边计算边作图. 从理论上已经知道,如果存在稳定周期轨,则上述轨将收敛于它, 这对应于分岔图上的低周期窗口. 在图 10-1 上可以清楚地看出的是周期 3 窗口.

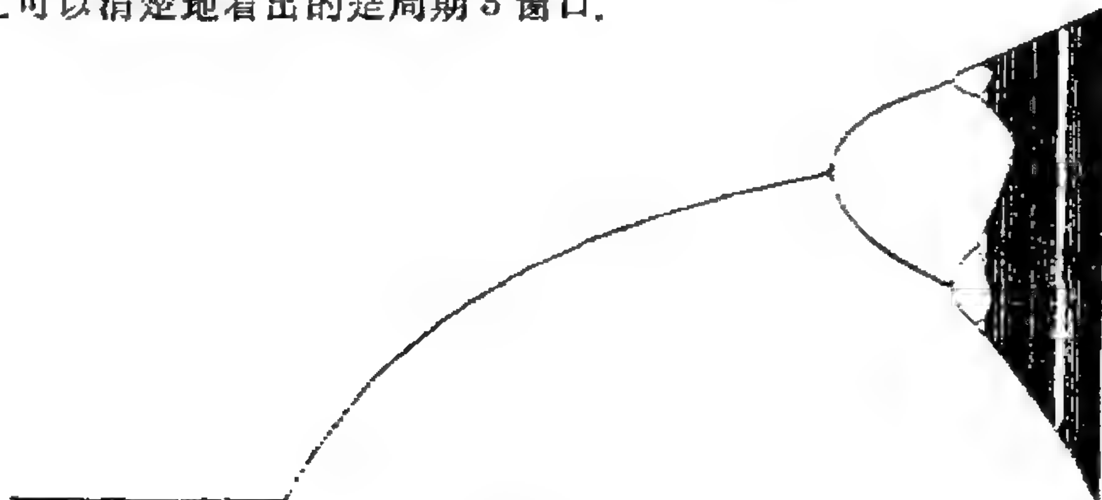


图 10-1

现在从 $b=0$ 开始观察. 当 $0\leqslant b\leqslant 1$ 时, 在分岔图上只有与 b 轴重合的一段直线. 它反映出 $x=0$ 为稳定不动点的事实. 从 $b=1$ 向右可以看到一条曲线,它对应于稳定不动点 $x=1-b^{-1}$, 直到 $b=3$ 为止.

从 $b=3$, $x=2/3$ 这个点向右, 我们看到曲线 $x=1-b^{-1}$ 发生分岔,变为两枝,它们对应于一个稳定周期 2 轨的出现. 通过简单的计算可知道, 在 $b>3$ 时的不动点 $x=1-b^{-1}$ 继续存在,但由于它不稳定,在上述计算机实验中是看不见的. 在图 10-1 上还可以看到二分为四和四分为八的分岔现象. 利用更为精细的计算机实验和理论研究可以知道, 这样的倍周期分岔过程在参数 b 增加时将不断继续下去. 同时与分岔对应的参数值很快收敛于一个极限值,记为 b_∞. 对二次方映射 $f(x)=bx(1-x)$, $b_\infty\approx3.5699456\cdots$. 我们称 b_∞ 为费根鲍姆点.

关于从倍周期分岔中所表现出来的普适性和在 $b=b_\infty$ 时的映射性质已有大量的研究. 读者可以从[20]~[29] 中找到有关材

料，还可以参考[56]～[59]．这里只指出几个主要性质．映射

$$f(x)=b_\infty x(1-x),\quad 0\leqslant x\leqslant 1,$$

存在所有周期为 2 的幂的周期轨，但不存任何其他周期轨．从临界点 0.5 出发的轨不是周期轨或终极周期轨，它的闭包是一个康托集，其勒贝格测度为 0，豪斯道夫维数（Hausdorff Dimension）为 0.538…（见[57]、[60]）．任何其他轨或者在有限步后直接落到某一个周期为 2 的幂的不稳定周期轨上，或者被吸引到上述康托集上．我们称这个康托集为**费根鲍姆吸引子**．由于这是系统在这时的主要特征，今后称具有这样的吸引子的单峰映射动力系统为"费根鲍姆吸引子（系统）"．

下面要确定费根鲍姆吸引子的揉序列．

§ 10.2 重正化变换与揉序列

考虑关于单峰映射的重正化变换（参见[22]，[24]）．观察图 10-2，其中含有一个单峰映射 f 的图象，c 为临界点．P 是在 c 右方的不动点，$\hat{P}$ 是满足条件 $f(\hat{P})=f(P)$ 的与 P 不同的另一个点．在 $\hat{P}\leqslant f^2(c)<c$ 的条件下，在图中作出了 f^2 在区间 $[\hat{P}, P]$ 上的图象，它与对角线有两个交点，除了 $P(=f^2(P))$ 之外的另一个交点即是 f 的一个周期 2 点．

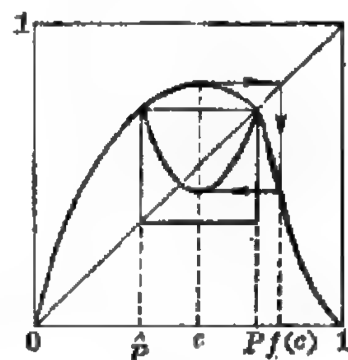


图　10-2

可以看出，如果将图 10-2 中以点 $(\hat{P}, \hat{P})$ 和点 (P, P) 为对顶点的正方形旋转一百八十度，再将区间 $[\hat{P}, P]$ 放大成区间 $[0, 1]$，则 f^2 的图象就成为与原来的 f 类似的单峰映射的图象了．在数学上可以将以上步骤用严格的表达式写出来，这里从略．我们称这样的变换为重正化变换．

由于在对于 f 作重正化变换后得到的仍为单峰映射，只要满足上面所说的条件，就可以再做同样的重正化变换．可以证明，对

$b=b_\infty$ 时的二次方映射，这个重正化变换过程可以无限制地进行下去．对于一般的单峰映射，我们就称可以无限次进行重正化变换的迭代映射系统为费根鲍姆吸引子．记这时的映射为 f，它的揉序列为

$$\mathrm{KS}(f)=a_1a_2\cdots a_n\cdots.$$

从图 10-2 可见，f 将$[\hat{P}, P]$映射到$[P, f(c)]$，又将$[P, f(c)]$映入$[\hat{P}, P]$．因此在轨

$$f^*(f(c))=(f(c), f^2(c), \cdots, f^n(c), \cdots)$$

中，当 n 为奇数时，总满足 $f^n(c)\in[P, f(c)]$．这表明，在揉序列 $\mathrm{KS}=A(f^*(f(c)))$中，当 n 为奇数时，总成立

$$a_n=A(f^n(c))=1.$$

于是 $a_1=a_3=a_5=\cdots=1$，即有

$$\mathrm{KS}(f)=1a_21a_41a_6\cdots a_{2n-2}1a_{2n}1\cdots.$$

从图 10-2 又可看出，在施行重正化变换之后，单峰映射 f^2 的揉序列为

$$\mathrm{KS}(f^2)=\bar{a}_2\bar{a}_4\bar{a}_6\cdots\bar{a}_{2n-2}\bar{a}_{2n}\cdots.$$

其中的上划"—"表示取反码，即 $\bar{0}=1$，$\bar{1}=0$．严格地说，对 f 作重正化变换后所得的单峰映射不能用 f^2 这个记号．为简明起见，我们约定上面的 f^2，包括在 $\mathrm{KS}(f^2)$中的 f^2，都是满足在 § 5.1 中条件的单峰映射．

根据条件，对 f^2 仍可施行重正化变换，因此根据同样理由，在 $\mathrm{KS}(f^2)$中的所有奇数位也都等于 1，即有

$$a_2=a_6=\cdots=a_{2+4k}=\cdots=0,\qquad k\geqslant 0,$$

于是得到

$$\mathrm{KS}(f^2)=1\bar{a}_41\bar{a}_81\bar{a}_{12}\cdots.$$

又从 $\mathrm{KS}(f^4)=a_4a_8a_{12}\cdots$可导出

$$a_4=a_{12}=\cdots=a_{4+8k}=\cdots=1,\qquad k\geqslant 0.$$

用数学归纳法不难求出 $\mathrm{KS}(f)=a_1a_2\cdots a_n\cdots$的每一项的表达式为

$$a_{2^k(2n+1)}=\begin{cases}1, & \text{当 } k \text{ 为偶数时;}\\ 0, & \text{当 } k \text{ 为奇数时,}\end{cases}$$

其中 $n\geqslant 0$.

由此可见，从 f 可以无限多次实施重正化变换出发，f 的揉序列是唯一确定的. 今后记这个揉序列为

$$t_\infty = a_1a_2\cdots a_i\cdots,$$

其中的 a_i 由上述公式给出.

下面采用另一种方法来研究上述序列 t_∞.

由于 f^2 与 f 一样，可以无限多次实施重正化变换，因此从上面的唯一性就知道有

$$\mathrm{KS}(f) = \mathrm{KS}(f^2),$$

或者写成

$$1a_21a_41a_61a_8\cdots 1a_{2n}\cdots = \bar{a}_2\bar{a}_4\bar{a}_6\bar{a}_8\cdots\bar{a}_{2n}\cdots.$$

由此可以看出，如果定义同态映射

$$h = \{0\to 11,\ 1\to 10\},$$

那么就得到 $h(\bar{a}_2\bar{a}_4\cdots) = 1a_21a_4\cdots$，即是

$$h(t_\infty) = t_\infty.$$

这里的同态映射概念在 §1.9 和 § 4.2 中已经引进. 按照 h 的定义，$h(\varepsilon) = \varepsilon$，以及对 S^* 中的任何串 x、y，令 $h(xy) = h(x)h(y)$，就可以将 h 应用到任何有限符号串或无限符号串上去.

现在从符号 1 出发，定义

$$t_0 = 1, \quad t_{n+1} = h(t_n), \qquad n\geqslant 0,$$

就得到序列 $\{t_n\}_n\geqslant 0$，其中 t_n 的长度为 2^n. 序列 $\{t_n\}$ 的前几个为

$$t_0 = 1, \qquad t_1 = 10,$$

$$t_2 = 1011, \qquad t_3 = 10111010.$$

如果用 $\hat{t}_n$ 记与 t_n 只有最后一个符号不同，而其他部分完全相同的符号串，那么用数学归纳法可以建立以下关系:

$$t_{n+1} = t_n\hat{t}_n, \qquad h(\hat{t}_n) = \hat{t}_{n+1},$$

$$h^n(1) = t_n, \quad h^n(0) = \hat{t}_n, \qquad n\geqslant 0.$$

其中规定对任何串 w，有 $h^0(w) = w$.

从关系式 $t_{n+1} = t_n\hat{t}_n$（其中 $n\geqslant 0$）可见，对任何 $i>n$，t_i 总以 t_n

为前缀. 因此在按位收敛意义上可以得到

$$t_\infty = \lim_{n\to\infty} t_n,$$

或记成 $t_\infty = h^\infty(1)$. 这样就可以将 t_∞ 定义为以任何 t_n 为前缀的一个无限符号串.

应用在 § 4.1 中的 DOL 系统, $\{t_n\}_{n\geq 0}$ 即是由 DOL 系统

$$G = (\{0, 1\}, \{0\to 11, 1\to 10\}, 1)$$

生成的 DOL 语言. 可以将 $\{t_n\}_{n\geq 0}$ 称为 **DOL 序列**, 而 t_∞ 即是这个序列的极限.

§ 10.3 t_∞ 与 TM 序列

为了研究语言 $\mathscr{L}(t_\infty)$, 即由探序列 t_∞ 确定的形式语言, 在这里先研究 t_∞ 的性质.

容易证明 t_∞ 不是周期序列或终极周期序列. 实际上, 如果可将它写成

$$t_\infty = yx^\infty,$$

其中 $|x| = p \geq 1$, 则可以将 p 以唯一的方式写为

$$p = 2^m(2n+1),$$

其中 m 和 n 是两个确定的整数. 取 k 充分大, 使 $k \geq m$, 同时满足

$$2^k(2n+1) \geq |y|.$$

我们观察在 $t_\infty = a_1 a_2 \cdots a_i \cdots$ 中的项

$$a_{2^k(2n+1)} \quad 与 \quad a_{2^{k+1}(2n+1)}.$$

由于它们的下标之差为 p 的倍数, 又都在 $t_\infty = yx^\infty$ 的周期后缀中, 因此应当相等. 但另一方面, 从 § 10.2 中的公式可见, 这两个符号一定是不相等的, 由此引出矛盾.

利用 § 8.3 的定理 1, 就知道费根鲍姆吸引子的形式语言 $\mathscr{L}(t_\infty)$ 不会是正规语言. 在[54]之前, 这一点已为许多作者猜测过, 见[44]、[48]、[49]等等.

在[54]中指出, 在 t_∞ 中不出现接连的四个符号 1, 也不出现 t_n^4 这样的子串, $n \geq 0$. 代替在[54]中的直接证明, 我们介绍在 t_∞

与著名的图厄-莫尔斯序列(Thue Morse Sequence)之间的联系，并由之可得到更强的结论.

对于 $S=\{0, 1\}$，定义同态

$$g=\{0\to 01,\ 1\to 10\},$$

并且从 $s_0=0$ 出发定义 $s_{n+1}=g(s_n)$，$n\geq 0$，就可以得到序列 $\{s_n\}_{n\geq 0}$. 其中的前几个为

$$s_0=0, \qquad s_1=01,$$

$$s_2=0110, \qquad s_3=01101001.$$

如果记 $\bar{s}_n$ 是将 s_n 中每个符号取反码所得到的序列，就可以建立递推关系

$$s_{n+1}=s_n\bar{s}_n, \qquad n\geq 0,$$

从而可以定义极限

$$s_\infty=\lim_{n\to\infty} s_n,$$

称 s_∞ 为**图厄-莫尔斯序列**，它是图厄(A. Thue)和莫尔斯分别独立发现的(参见[61]、[31]、[32]、[4]等). 图厄的出发点是寻找不含任何重复子串(即子串平方)的无限符号序列. 莫尔斯则是为了研究负曲率曲面上的测地线性态，但后来也研究了与图厄相同的问题(见[62]).

将一个在其中不出现任何形式为 $x^2(=xx)$ 的子串的无限符号序列称为**无平方串**(Square-Free)**序列**. 同样，可定义**无立方串**(Cube-Free)**序列**. 介于二者之间还有一种**强无立方串**(Strongly Cube-Free)**序列**，即不允许出现任何形式为 x^2a 的子串，其中 a 是串 x 的第一个符号.

仅仅用两个符号不能构造出无平方串的无限序列，但是可以证明

定理 s_∞ 是强无立方串的无限序列.

在这个基础上，可以用三个符号构造出无平方串的无限序列. 关于这些内容的详细讨论可参看[4]、[61]、[62]，此处从略.

现在叙述在 s_∞ 与 t_∞ 之间的联系. 记 $t_\infty=a_1a_2\cdots a_i\cdots$，$s_\infty$

$=b_0b_1\cdots b_i\cdots$. 令 $b_0=0$, 并且令

$$b_{i+1}=\begin{cases}b_i, & 如\ a_{i+1}=0;\\ 1-b_i, & 如\ a_{i+1}=1,\end{cases}$$

$i\geqslant 0$, 就可以从 t_∞ 得到 s_∞. 利用 s_∞ 为强无立方串序列, 就直接导出我们所需要的结果:

推论 t_∞ 是无偶平方串的序列.

实际上, 如 t_∞ 含子串 y^2, y 为偶串, 则利用上述联系, 就会在 s_∞ 的相应处找到形状为 x^2a 的子串, 其中 a 为 x 的第一个符号, 由此引出与上述定理相矛盾.

由这个推论立即知道, 在 t_∞ 中不会出现 1^4 或 t_n^4 这样的子串了. 容易知道, 对每个 n, t_n 是奇串, 但 $t_n^4=(t_n^2)^2$, 而 t_n^2 当然是偶串.

在文献[63]、[64]中已经指出了在 s_∞ 与 t_∞ 之间的联系. 此外, 在费根鲍姆的研究之前, 序列 t_∞ 已在[65]、[34]中出现.

关于图厄-莫尔斯序列的研究和应用, 还可参看[66].

§10.4 语言 $\mathscr{L}(t_\infty)$ 的结构

对于非正规语言 $\mathscr{L}(t_\infty)$, 我们先求出由揉序列 t_∞ 决定的所有允许字, 然后就可以得到语言 $\mathscr{L}(t_\infty)$ 中所有的字.

在确定那些是允许字时, 主要依赖于下列的基本结果.

命题 如果 s 是以 t_∞ 为揉序列时的允许字, s 以某个 $t_k(k\geqslant 0)$ 为前缀, 即有 $s=t_kv$, 那么 v 的长度为 $2^k(=|t_k|)$ 的前缀只有两种可能: (i), v 以 t_k 为前缀, (ii), v 以 $\hat{t}_k$ 为前缀.

这个命题实际上完全确定了允许字的所有可能构造, 其证明写在书末的附录 C 中.

现在叙述这个命题的应用.

从 s 的第一个符号开始, 如果 $s=0^\infty$, 它已是一个允许字. 否则, s 中含有符号 1, 可以写成

$$s=0^{n_{-1}}s',$$

其中 s' 从符号 1 开始, $n_{-1}\geqslant 0$ 代表在 s' 之前的 0 串长度. 从允许

字定义可知，s' 作为 s 的后缀也是允许字．对 s' 和 $t_0=1$ 用命题．如果在这个 t_0 之后仍出现 t_0，则对这个 t_0 可以再用命题．如在 t_0 之后出现 $\hat{t}_0(=\hat{1}=0)$，则合并 $t_0\hat{t}_0$ 为 t_1，写成

$$s=0^{n-1}t_1s'',$$

其中 $t_1s''=s'$．对 s' 和 t_1 再用命题．这样继续下去有以下三种可能性：

(i) 从某个 t_l 开始，每次应用命题时始终只出现相同的 t_l（而不出现 $\hat{t}_l$）．这样就得到

$$s=0^{n-1}t_0^{n_0}t_1^{n_1}\cdots t_{l-1}^{n_{l-1}}t_l^{\infty}.$$

对于满足施瓦兹导数为负条件的单峰映射，这样的允许字对应于在§10.1 中所说的在有限步后直接落到一个周期为 2 的幂的不稳定周期轨上的情况．实际上，每个 $|t_l|=2^l$，t_l^{∞} 本身也是允许字．应用§6.3 的定理 2，上述不稳定周期轨的踪迹就是 t_l^{∞}．我们知道，这时不存在任何其他类型的周期轨．

(ii) 对每一个 t_l 在应用命题时，在有限次出现 t_l 之后一定会出现 $\hat{t}_l$，这时可得到 $t_l\hat{t}_l=t_{l+1}$．同时，在命题中的第二种情况也不会无限制地继续发生．这样就得到

$$s=0^{n-1}t_0^{n_0}t_1^{n_1}\cdots t_l^{n_l}\cdots,$$

每个 $n_l\geqslant 0$．这也可记为乘积形式

$$s=0^{n-1}\prod_{l=0}^{\infty}t_l^{n_l}.$$

由于揉序列 t_∞ 以任何 t_l 为前缀，这样的允许字反映出轨趋向于费根鲍姆吸引子的动力学行为．

(iii) 从某个 t_l 开始，每次应用命题时只出现第二种可能性．这样一来，利用 $t_l\hat{t}_l=t_{l+1}$，$t_{l+1}\hat{t}_{l+1}=t_{l+2}$，…，就要到

$$s=0^{n-1}t_0^{n_0}\cdots t_l^{n_l}t_\infty.$$

这时 s 的某个移位与揉序列相等．

由于已知 t_∞ 不是周期序列，应用在§6.5 中的允许字定义和定理 3，可以不考虑这类允许字．实际上这类允许字未必反映了

动力系统的真实轨的行为，这与图 5·1 的讨论类似.

最后，取上述前两类允许字的所有前缀，就可以得到所要求的形式语言 $\mathscr{L}(t_\infty)$. 为方便起见，引入规范串的概念.

称具有下列形式的符号串为规范串：

$$t_{-1}^{n_{-1}} t_0^{n_0} t_1^{n_1} \cdots t_l^{n_l},$$

其中记号 $t_{-1}=0$，指数 $n_i(-1\leqslant i\leqslant l)$ 为非负整数. 如果每个指数为 0，即代表空串 ε. 否则，设 $n_l\geqslant 1$ 成立.

又引入记号 $p(t_k)$ 表示串 t_k 的任何真前缀.

这样就可以写出

$$\mathscr{L}(t_\infty)=\{t\in S^* \mid t=up(t_k),\ u\text{为某个规范串},\ k\geqslant 0\}.$$

实际上，从两类允许字取前缀时，所得的形式是相同的. 此外，可以看出，在 $t=up(t_k)$ 中，u 的规范串形式中的 l 与 k 无关.

§ 11 复杂性分析

在上一节的基础上，用奥登引理（见§ 3.4）证明 $\mathscr{L}(t_\infty)$ 不是上下文无关语言. 又应用在第 4 节中介绍的并行重写系统，证明 $\mathscr{L}(t_\infty)$ 是 ETOL 语言，从而也是上下文有关语言. 本节的材料取自[54].

§ 11.1 关于 t_n 的一些性质

在下面讨论中需要关于序列 $\{t_n\}_{n\geqslant 0}$ 的一些有关性质.

在附录 B.6 中引进了关于一个有限符号串的循环移位最大串的概念，并将串 x 的循环移位最大串记为 $M(x)$

利用 t_n 为奇串，$\hat{t}_n$ 为偶串 $(n\geqslant 0)$，容易证明每个 t_n 都是循环移位最大串，即

$$t_n=M(t_n), \qquad n\geqslant 0.$$

还可以证明

$$0^\infty < t_0^\infty < t_1^\infty < \cdots < t_n^\infty < t_{n+1}^\infty < \cdots < t_\infty.$$

每一个 t_n^∞ 都是 $KS=t_\infty$ 时的允许字. 按照§6.3的定理2, 它们一定是周期轨的踪迹. 容易证明 t_n^∞ 的最小周期等于 t_n 的长度 2^n. 由于 t_n 是奇串, 按照上述定理, 还可能存在最小周期为 $2|t_n|$ 的周期轨, 也以 t_n^∞ 为踪迹.

同时, 每个 t_n^∞ 都是移位最大字, 因此都可以是某个单峰映射的揉序列. 这样就可以将上面的一系列不等式看成为揉序列之间的关系. 实际上, 这反映了倍周期分岔的过程. 重新观察图 10-1. 当参数 b 的值在$[0, 2)$时, 揉序列为 0^∞. 当 b 在$(2, 1+\sqrt{5})$中时, 揉序列为 1^∞, 而在其中的 $b=3$ 处发生倍周期分岔. 在 $b=1+\sqrt{5}$时, 揉序列为$(1c)^\infty$. b 再增大时, 揉序列变为$(10)^\infty$, 即 t_1^∞. 以后的情况与此类似. 也就是说, 当 b 增长到某个值之后, 揉序列变为 t_n^∞, 这时存在以它为踪迹的稳定周期轨, 最小周期为 $|t_n|(=2^n)$. 当参数 b 继续增大到某个值之后, 这个周期轨变为不稳定, 同时出现了周期加倍的稳定周期轨. 它们的踪迹均为 t_n^∞. 然后当 b 值增大达到某个值时, 出现超稳定周期轨 $t_n|_c$, 在这之后出现揉序列 t_{n+1}^∞.

在费根鲍姆点 b_∞ 处, 所有这些以 t_n^∞ 为踪迹的周期轨都同时存在, 但都是不稳定的. 可以在理论上直接证明, 当揉序列为 t_∞ 时, 不存在任何踪迹不是某个 t_i^∞ 的周期轨. 而当揉序列为 $t_n^\infty(n\geqslant 0)$ 时, 不存在任何踪迹不是 t_i^∞, $i\leqslant n$ 的周期轨. 这个结论比不存在周期不是 2 的幂的周期轨还要强.

以上的一些事实可于[24]等文献中找到, 也不难直接证明, 这里从略.

§11.2 $\mathscr{L}(t_\infty)$ 不是 CFL 的证明

现在应用在§3.4 介绍的工具来证明语言 $\mathscr{L}(t_\infty)$不是上下文无关的. 为了不涉及到过多的技术细节, 这里只对证明的主要过程作一个介绍.

从上一节关于允许字的命题可以知道, 如果在一个字 $z\in$

$\mathscr{L}(t_\infty)$中出现一个子串 t_k，则在这之后下标 p 低于 k 的子串 t_p 的出现会受到某种限制．利用在 § 10.3 中的推论，我们知道在 t_∞ 中不出现任何 $t_n^4(n\geqslant 0)$，这样就可以将上述想法严格化，得到

命题　如 $z\in\mathscr{L}(t_\infty)$，$z=t_k v t_p^n$，$p<k$，那么 $n\leqslant 3$．

这个命题的证明并不困难，这里从略．

现在来证明 $\mathscr{L}(t_\infty)$不是上下文无关语言．

用反证法．如果 $\mathscr{L}(t_\infty)$是上下文无关语言，则由 § 3.4 中的奥登引理，存在一个只与语言有关的整数 n，满足该引理中的要求．

取 k，使 $2^{k-1}>n$ 成立．然后取字

$$z=t_k^4\in\mathscr{L}(t_\infty).$$

利用关系式 $t_k=t_{k-1}\hat{t}_{k-1}$，可将 z 改写为

$$z=t_k^3 t_{k-1}\hat{t}_{k-1},$$

然后按引理中的规定，我们将 z 的最后 2^{k-1} 个符号的位置选定为可区分位．从引理的叙述可知，字 z 和可区分位都是可以任意取的，只要 z 的长度大于 n 和可区分位的个数大于 n．

根据引理，上述字 z 可分解为

$$z=uvwxy,$$

且满足以下条件：

1° 或者 u 和 v 均含可区分位，或者 x 和 y 均含可区分位，二者至少有一个成立．

2° w 至少含一个可区分位．

3° vwx 至多含 n 个可区分位．

4° 对每个 $i\geqslant 0$，$uv^iwx^iy\in\mathscr{L}(t_\infty)$．

可以证明，不可能存在这样的分解．

如果 $x\neq\varepsilon$，则由于 w 含可区分位，而 z 的可区分位是 z 的一个后缀，因此 x 的每一位都是可区分位．同时有 $|x|\leqslant 2^{k-1}$ 成立．

从条件 4° 推出对每个 $i\geqslant 0$ 有 $x^i\in$ L，因此 x^ω 为允许字．利用在 § 11.1 中关于周期允许字的结论，可见 x 是某个 t_p 或它的幂，其中 $p<k-1$，最多差一个循环移位．但从

$$z = t_k^{\hat{4}} = t_k t_k t_k t_{k-1} \hat{t}_{k-1} = uvwxy$$

可以看出，由于 w 至少含一个可区分位，而 z 的可区分位集中在后缀 $\hat{t}_{k-1}$ 中，因此在 u、v、w 这三个子串中，至少有一个子串会含有一个完整的子串 t_k. 现在利用

$$uv^i wx^i y \in \mathscr{L}(t_\infty),$$

而其中的 i 可以任意大，就导致与本小节的命题相矛盾.

如果 $x=\varepsilon$，则从条件 1° 知道，这时 u 和 v 同时含有可区分位，因此 $|v|<2^{k-1}$，而 u 含有 t_k. 以下证明过程与 $x\neq\varepsilon$ 的情况完全相同.

§ 11.3 $\mathscr{L}(t_\infty)$ 为 ETOL 语言的证明

利用在 § 4.2 中的 ETOL 语言，或者在 § 4.5 中的标号语言，都可以证明 $\mathscr{L}(t_\infty)$ 为上下文有关语言（参看在 § 4.3 中的讨论）. 由于 ETOL 语言类为标号语言的真子类，即有

$$\mathscr{L}(\text{ETOL}) \subsetneq \mathscr{L}(\text{IND}) \subsetneq \mathscr{L}(\text{CS}),$$

我们在这里将证明 $\mathscr{L}(t_\infty)$ 为 ETOL 语言，从而比较准确地决定它的语言复杂程度.

利用 § 10 中关于语言 $\mathscr{L}(t_\infty)$ 的结构分析，将它写成三个语言的连接(Concatenation)是比较方便的:

$$\mathscr{L}(t_\infty) = K_1 K_2 K_3,$$

其中

$$K_1 = \{0^n \mid n \geqslant 0\},$$
$$K_2 = \{u \mid u \text{ 为规范串 } t_0^{n_0} t_1^{n_1} \cdots t_i^{n_i}\},$$
$$K_3 = \{v \mid v = p(t_k),\ t_k \geqslant 0\}.$$

K_1 是正规语言，也是 OL 语言，当然也是 ETOL 语言. K_2 是最主要的部分. 关于规范串的定义以及在 K_3 中的记号 $p(t_k)$ 均见 § 10.4.

定义一个 ETOL 系统为

$$G_2(\{s_0,\ T,\ 0,\ 1\},\ H,\ s_0,\ S),$$

其中 $S=\{0,\ 1\}$，$H=\{h_1,\ h_2\}$，

$$h_1=\{s_0\to \varepsilon \mid s_0 \mid s_0T,\ T\to T,\ 0\to 0,\ 1\to 1\},$$
$$h_2=\{s_0\to s_0,\ T\to 1,\ 0\to 11,\ 1\to 10\}.$$

记 $U(G_2)$ 为 G_2 的基础系统，即 TOL 系统

$$U(G_2)=(\{s_0,\ T,\ 0,\ 1\},\ H,\ s_0).$$

引理 1 由 $U(G_2)$ 生成的语言为

$$\mathrm{L}(U(G_2))=\{T^n u \mid u=t_0^{n_0}t_1^{n_1}\cdots t_l^{n_l},\ n\geqslant 0\}$$
$$\cup\{s_0T^n u \mid u=t_0^{n_0}t_1^{n_1}\cdots t_l^{n_l},\ n\geqslant 0\}.$$

证明 先证等号右方为左方的子集. 如 $u=\varepsilon$，则容易直接看出 T^n 和 s_0T^n 属于 $\mathrm{L}(U(G_2))$. 在 $u\neq\varepsilon$ 时，对 l 用数学归纳法.

在 $l=0$ 时，从

$$s_0\overset{1}{\Rightarrow} s_0T^{n_0}\overset{2}{\Rightarrow} s_0t_0^{n_0}\overset{1}{\Rightarrow} s_0T^nt_0^{n_0}\overset{1}{\Rightarrow} T^nt_0^{n_0}$$

可见成立. 这里 n 为任何非负整数，记号“$\overset{1}{\Rightarrow}$”表示只利用 h_1 进行重写，“$\overset{2}{\Rightarrow}$”表示只利用 h_2 进行重写，次数不限.

现设对 l 已成立，考虑 $l+1$ 的情况. 这时可以从

$$
\begin{aligned}
s_0 &\Rightarrow s_0T^{n_0}t_0^{n_1}\cdots t_l^{n_{l+1}}\\
&\overset{2}{\Rightarrow} s_0t_0^{n_0}t_1^{n_1}\cdots t_{l+1}^{n_{l+1}}\\
&\overset{1}{\Rightarrow} s_0T^nt_0^{n_0}t_1^{n_1}\cdots t_{l+1}^{n_{l+1}}\\
&\overset{1}{\Rightarrow} T^nt_0^{n_0}t_1^{n_1}\cdots t_{l+1}^{n_{l+1}}
\end{aligned}
$$

得到证明. 其中第一步的“$\Rightarrow$”即使用在 l 时已成立的归纳假设.

现在证明引理的后一半，即如有 $z\in\mathrm{L}(U(G_2))$，则 z 一定具有形式 T^nu 或 s_0T^nu，其中 $n\geqslant 0$，u 为规范串 $t_0^{n_0}t_1^{n_1}\cdots t_l^{n_l}$.

用记号 $s_0\Rightarrow_{n'}z$ 表示从 s_0 出发，经过 n' 次重写得到 z. 对 n' 用数学归纳法.

对于 $n'=0$ 或 $n'=1$，这是平凡的. 现设上述论断对 n' 已成立，考虑 $n'+1$ 的情况.

设 $s_0\Rightarrow_{n'}z=T^nu$，$u=t_0^{n_0}t_1^{n_1}\cdots t_l^{n_l}$. 再重写一次，如用 h_1，则 $h_1(z)=\{z\}$. 如用 h_2，则 $h_2(z)=\{t_0^{n}t_1^{n_0}\cdots t_{l+1}^{n_l}\}$.

设 $s_0\Rightarrow_{n'}z=s_0T^nu$，那么有

$$h_1(x)=\{T^n u,\ s_0T^n u,\ s_0T^{n+1}u\},$$

$$h_2(x)=\{s_0t_0^{n_1}t_1^{n_0}\cdots t_{l+1}^{n_l}\}.$$

这样就证明了在重写 $n'+1$ 次时所得到的符号串仍具有所说的形状. 证毕.

根据 ETOL 语言的定义, 就有

$$\mathrm{L}(G_2)=\mathrm{L}(U(G_2))\cap S^*=K_2.$$

这样就已证明了 K_2 为 ETOL 语言.

再讨论语言 K_3. 构造一个 EOL 系统

$$G_3=(\{B,\ 0,\ 1\},\ h,\ B,\ S),$$

其中 $S=\{0,\ 1\}$,

$$h=\{B\to \varepsilon|B|1|1B,\ 0\to 11,\ 1\to 10\}.$$

同样先研究 G_3 的基础系统 $U(G_3)$, 它是 OL 系统($\{B,\ 0,\ 1\},\ h,\ B$).

引理 2 $\mathrm{L}(U(G_3))=\{v|v=p(t_k),\ k\geqslant0\}\cup\{vB|v=p(t_k),\ k\geqslant0\}$.

证明 对 k 用数学归纳法, 可以证明右方为左方的子集. 为了方便起见, 我们证明更强一些的结论, 即从 B 出发可以恰好经过 k 次重写得到 $p(t_k)$ 或 $p(t_k)B$, 记为

$$B\Rightarrow_k p(t_k)\quad 与\quad B\Rightarrow_k p(t_k)B.$$

对于 $k=1$, 从 $t_1=10$ 可见, 作为 t_1 的真前缀, $p(t_1)$ 只有 1 和 ε 两种可能. 直接可以写出

$$B\Rightarrow_1\varepsilon,\ B\Rightarrow_1 1,\ B\Rightarrow_1 B,\ B\Rightarrow_1 1B.$$

现设上述论断对 k 已成立, 讨论 $k+1$ 的情况. 先看 $v=p(t_{k+1})$. 从 $t_{k+1}=t_k\hat{t}_k$ 可见, 如果 v 的长度小于 $|t_k|(=2^k)$, 则 v 也是 t_k 的真前缀. 利用 $B\Rightarrow_1 B\Rightarrow_k v$, 可见 $B\Rightarrow_{k+1}v$ 成立. 如果 $|v|\geqslant|t_k|$, 则可以记 $v=t_kv'$, v' 也是 t_k 的真前缀. 这里利用了 t_k 与 $\hat{t}_k$ 仅在最后一个符号上不同. 于是从 $B\Rightarrow_1 1B\Rightarrow_k t_kv'=v$ 得到 $B\Rightarrow_{k+1}v$.

关于 $v=p(t_{k+1})B$ 的讨论相同, 故从略.

现在证明引理 2 的后一半，即 $L(U(G_3))$ 中每个字的形式不是 $p(t_k)$，便是 $p(t_k)B$，$k\geqslant 0$.

设 $B\Rightarrow_n x$，对 n 用数学归纳法.

对 $n=1$ 这是平凡的.现设结论对 n 已成立，讨论 $n+1$ 的情况.

如有 $B\Rightarrow_n p(t_k)$，则再重写一次后的 $h(p(t_k))$ 明显为某一个 $p(t_{k+1})$.

如有 $B\Rightarrow_n p(t_k)B$，则可以计算出再重写一次后的所有可能为

$$h(p(t_k)B)=\{p(t_{k+1}),\ p(t_{k+1})B,\ h(p(t_k))1,\ h(p(t_k))1B\}.$$

由于 $h(p(t_k))1$ 也是 t_{k+1} 的一个真前缀，归纳已经完成. 证毕.

根据 EOL 语言的定义，我们有

$$L(G_3)=L(U(G_3))\cap S^*=K_3.$$

于是，已知 K_3 为 EOL 语言，当然也是 ETOL 语言.

最后，利用语言类 $\mathscr{L}$(ETOL) 对于连接运算封闭，就证明了 $\mathscr{L}(t_\infty)$ 为 ETOL 语言，从而也是标号语言和上下文有关语言.

§11.4 讨 论

在完成[54]之后，我们看到了论文[44]，在其中已研究了类似的问题. 在[44]中，从揉序列 t_∞ 出发取它的所有有限子串来定义一个形式语言 L. 然后用关于上下文无关语言的泵引理(见 §3.4)证明 L 不是上下文无关语言，用构造标号语法的方法证明 L 是标号语言，因而也是上下文有关语言. 这里的讨论要容易得多. 但在[44]中的语言 L 只反映了吸引子自身中的符号动力学行为.

在[48]中对于费根鲍姆吸引子宣布了类似于[44]的结果，但到目前为止尚未见到其证明，也不了解所讨论的形式语言是什么.

仿照 §3.6 中的例子对语言 $\mathscr{L}(t_\infty)$ 直接写出上下文有关语法是可以做到的，但是证明将会变得相当冗长. 从 §4.4 中关于 $\mathscr{L}$(ETOL) 的介绍可以看到，它在上下文有关语言类 $\mathscr{L}$(CS) 中是复杂程度相对较低的一个子类.

此外，还应指出的是，在[42]中对 $\mathscr{L}(t_\infty)$ 已作出了一个无限自

动机，如图 11-1 所示．它具有无限多个状态，即是在本书 § 2.1 中讨论过的那种自动机．一般而言，只要 KS 不是周期序列或终极周期序列，应用 § 8.7 中的定理 4，揉序列 KS 的无穷多个前缀分属于不同的 R_L 等价类，应用在迈希尔-奈罗德定理证明中的方法，就可以构造出如图 11-1 那样的无限自动机．但是对于一般的非正规语言，这类自动机有可能具有非常复杂的结构．



图 11-1

§ 12 其他非正规语言

本节先介绍上一节的直接推广，即在更为一般的倍周期分岔和 n 周期分岔的极限情况找到了新的非正规语言．然后介绍由斐波那契序列方法所生成的非正规语言．最后讨论在单峰映射的形式语言研究中有待解决的几个问题．材料来源为[47]～[49]与尚未发表的工作[55]、[67]等．

§ 12.1 关于 $\mathscr{L}(t_\infty)$ 的推广

在[55]中先研究了在每一个周期窗口中的倍周期分岔的极限．图 12-1 是二次方映射 $f(x)=bx(1-x)$ 当 b 从费根鲍姆点 b_∞ 增大到 4 时的分岔图．图中可看出三个较宽的周期窗口．左边是周期 6 窗，它的左端对应于揉序列 $(101110)^\infty$．中间是周期 5 窗，它的左端对应于揉序列 $(10111)^\infty$．右边是周期 3 窗，它的左端对应于揉序列 $(101)^\infty$．

由于每一个周期窗的左端都是由切分岔而形成的，因此容易知道，对应的揉序列如记为 x^∞，则 x 一定是偶串（参见[24]、[26]）．当参数 b 增大时，揉序列将依次变为 $x|_o$(即将 x 最后一个

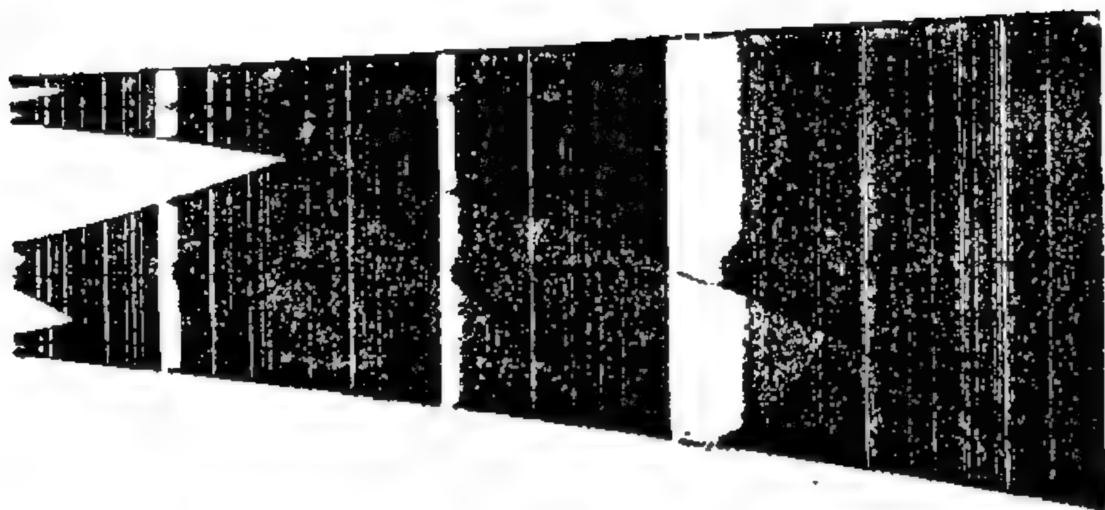


图 12-1

符号改为 o)，$\hat{x}^{\infty}$ 等等. 引进同态

$$h_x=\{0\to x,\ 1\to \hat{x}\},$$

则由 § 9.5 中提到的 * 合成律可知，在这个窗口中的所有揉序列从左到右的排列即是将 h_x 作用到从 0^{∞} 到 10^{∞} 的全部揉序列的结果，即自相似性. 特别是在窗右端的激变由揉序列

$$h_x(10^{\infty})=\hat{x}x^{\infty}$$

所刻划. 对周期 3 窗的激变见 § 8.6 的讨论.

容易看出，在每个周期窗中的倍周期分岔的极限，可以由揉序列 $h_x(t_{\infty})$ 来刻划. 我们称对应的动力系统为广义费根鲍姆吸引子.

在 [55] 中证明了，广义费根鲍姆吸引子的形式语言 $\mathscr{L}(h_x(t_{\infty}))$ 的复杂性和 $\mathscr{L}(t_{\infty})$ 相同，即不是上下文无关语言，是 ETOL语言，从而也是上下文有关语言.

在[55]中进一步考虑了 n 周期分岔的极限点. 例如，在图 12-1 的周期 3 窗中可以看到三个较小的图形，它们中的每一个与二次方映射的整个分岔图是相似的. 因此在这三个小图形中各有一个周期 3 窗，对应于原有的二次方映射即是周期 9 窗. 依次类推，就可以得到三周期分岔的极限点. 它的揉序列用同态来表示同样是非常方便的. 如像上面那样，记一个周期窗对应的同态为 h_x，那么令

$$w_n = h_\alpha^n(1), \qquad n \geqslant 0.$$

就得到极限点的揉序列为

$$w_\infty = \lim_{n\to\infty} w_n,$$

或记为

$$w_\infty = h_\alpha^\infty(1).$$

对于语言 $\mathscr{L}(w_\infty)$的讨论要比前面困难一些，这里在一些具体问题上需要建立某些新的方法. 在[55]中证明了 $\mathscr{L}(w_\infty)$不是上下文无关语言，而是 ETOL 语言，从而也是上下文有关语言.

§12.2 斐波那契系统

在[49]中指出，除了在倍周期分岔的极限点处有可能找到非正规语言之外，还可能从斐波那契序列的极限来寻找非正规语言，并作了大量的数值实验. 在[67]中对此进行了理论分析，发现了大量的非正规语言，并称相应的动力系统为**斐波那契系统**.

先举一个例子.

考虑同态

$$h = \{0 \to 1,\ 1 \to 10\}.$$

从 $t_0 = 1$ 开始，令 $t_n = h^n(t_0)$，$n \geqslant 1$，就得到一个(DOL)序列 $\{t_n\}_{n\geqslant 0}$. 它的前几个是

$$1,\ 10,\ 101,\ 10110,\ 10110101,\ \cdots$$

它们的长度即是斐波那契数列: 1, 2, 3, 5, 8, $\cdots$. 因此我们称 $\{t_n\}_{n\geqslant 0}$ 为**斐波那契序列**.

但这时的 t_n 不一定是循环移位最大字(参见附录 B.5). 在上面写出的第五个串就不是循环移位最大字. 按照[49]，将 t_n 循环移位，求出其中最大的 $m_n = M(t_n)$，就可以得到极限

$$m_\infty = \lim_{n\to\infty} m_n.$$

它是移位最大字，即可以作为某个单峰映射的揉序列. 在[49]中用数值方法求出了映射族 $x \mapsto \mu - x^2$ 中，当 $\mu = 1.714744850\cdots$时以 m_∞ 为其揉序列. 这个映射族与本书中的二次方映射 $f(x)$

$=bx(1-x)$等价，上述参数值换算成这里的 $b=3.80338713\cdots$。

除了用同态方法外，[49]中还使用了符号串置换(Block Substitution)方法来生成新的揉序列(还可以参看[45]、[63])。

从任意两个初始符号串 t_0、$t_1(\in S^*)$出发，按照规则

$$t_{2n}=t_{2n-2}t_{2n-1},\quad t_{2n+1}=t_{2n}t_{2n-1}$$

可以得到$\{t_n\}_{n\geq 0}$. 同样求出 $m_n=M(t_n)$，$n\geq 0$，并令 $n\to\infty$，就可能生成新的揉序列.

在[49]中考虑了$|t_0|=1$和$|t_1|=2$的所有六种可能情况，得到四种不同的序列$\{m_n\}_{n\geq 0}$，即得到了四个新的揉序列. 在表 12.1 中列出了这四个揉序列在二次方映射中对应的参数值. 与上面的例子一样，这里的参数值均已换算成在本书的二次方映射 $f(x)=bx(1-x)$中的参数值.

表 12.1

	t_0	t_1	b
1	0	11	3.800284948…
2	1 1	01 10	3.80338713…
3	0 0	10 01	3.904142834…
4	1	00	3.99251571…

在表 12.1 中的第二种情况与上面的例子用同态得到的结果相同.

在[67]中研究了较一般的情况. 称$\{t_n\}_{n\geq 0}$为(广义)斐波那契序列，如果从给定的初始串 t_0 和 t_1 出发，按以下四种方式中的任何一种生成序列$\{t_n\}_{n\geq 0}$:

A. $t_{n+2}=t_{n+1}t_n$,

B. $t_{n+2}=t_nt_{n+1}$,

C. $t_{2n+2}=t_{2n}t_{2n+1}$, $t_{2n+3}=t_{2n+2}t_{2n+1}$,

D. $t_{2n+2}=t_{2n+1}t_{2n}$, $t_{2n+3}=t_{2n+1}t_{2n+2}$.

在表 12.1 中所用的是方式 C.

已经证明，这四种方式在生成揉序列时的能力完全相同. 同时,不论如何取 t_0 和 t_1，序列 $m_n=M(t_n)$ 在 $n\to\infty$ 时一定收敛于某个移位最大字,即揉序列.

在这里需要区别三类不同情况.

1. $t_0t_1=t_1t_0$. 从附录 B.3 可知，每个 t_n 和 $m_n=M(t_n)$ 都是非既约串，揉序列一定是周期序列. 因此这时不会产生出非正规语言. 这相当于在表 12.2 中不予考虑的情况: $t_0=0$, $t_1=00$ 和 $t_0=1$, $t_1=11$.

2. $t_0t_1\neq t_1t_0$, t_0 和 t_1 均为偶串. 这时每个 t_n 与 $m_n=M(t_n)$ 都是偶串,今后称为**偶斐波那契序列**.

3. $t_0t_1\neq t_1t_0$, t_0 和 t_1 中至少有一个为奇串. 这时 t_n 和 $m_n=M(t_n)$ 的奇偶性按照两奇一偶的规律变化，今后称为**奇斐波那契序列**. 实际上,在接连的三个串中总有两个奇,一个偶.

已经证明，第 2、3 类序列所得到的揉序列一定不是周期序列或终极周期序列，因而就得到了新的非正规语言. 我们称相应的动力系统为**偶斐波那契系统**和**奇斐波那契系统**.

§12.3 关于同态的几个例子

已经证明，斐波那契系统的揉序列与本章中所讨论的其他情况一样,都可以用同态的方法生成,但这里可能涉及到符号串的更一般的置换.

以表 12.1 中的四个情况为例来说明.

[例 1] 从 $t_0=0$ 和 $t_1=11$ 出发按方式 C 生成的揉序列，可以用关于两个符号 α 和 β 的同态

$$h=\{\alpha\to\alpha\alpha\beta,\quad \beta\to\alpha\beta\}$$

得到，其中 $\alpha=101$, $\beta=11$. 从 101 或 11 出发,不断应用 h,就得

到揉序列的越来越长的前缀，其极限即是揉序列．这是偶斐波那契系统，而其他三个例子都是奇斐波那契系统．

[例 2]　从 $t_0=1$ 和 $t_1=01$(或 10)出发，按方式 C 生成揉序列．如上一小节中所看到的那样，也可以用同态{0→1, 1→10} 得到相同的揉序列．但是这两种方法都要求先作循环移位，再取极限．实际上，在[67]中证明，只要用同态

$$\{1\to10110,\ 0\to110\}$$

就可以从 1(或 0)出发直接生成相同的揉序列．

[例 3]　从 $t_0=0$ 和 $t_1=10$(或 01)出发，与例 2 类似，只要用同态

$$\{1\to100,\ 0\to10100\}$$

就可直接生成在表 12·2 中所对应的揉序列．

[例 4]　从 $t_0=1$ 和 $t_1=00$ 出发，则可以将例 3 中的揉序列再用{1→1, 0→00}作用上去，就得到表 12·2 中所对应的揉序列．或者也可以将

$$\{1\to10000,\quad 00\to10010000\}$$

直接用到符号 1 无限多次而得到相同的结果．

在以上四种情况得到的揉序列关于相应的同态都是不变的，即是同态的不动点．这对于一般情况也已证明成立．

在表 12·2 中写出了四个例子的揉序列的前 32 位．只要再将相应的同态作用上去，就可以得到揉序列的任意长度的前缀．对于例 1，在[45]中也有讨论．

表　12.2

1	10110, 11110, 11011, 11011, 11011, 01111, 01
2	10110, 11010, 11010, 11011, 01011, 01011, 01
3	10010, 10010, 10010, 01010, 01001, 01001, 01
4	10000, 10010, 00010, 01000, 01000, 01001, 00

§12.4 有待解决的问题

在这一小节中我们将对于单峰映射的形式语言作一般性的讨论，并提出几个有待解决的问题.

以乔姆斯基的层次结构理论来研究单峰映射的形式语言的复杂性问题，可以说只有正规语言方面的情况已经清楚，而在其他方面的研究成果还是很少的.

在[54]中提出了一个猜测，即在$\mathscr{L}$(KS)中不存在是上下文无关语言而同时不是正规语言的例子. 实际上，费根鲍姆吸引子的动力学行为相对来说还是比较简单的，它处在规则运动与不规则运动的临界点上. 但从第 11 节的研究可见，它的形式语言复杂性已处在高于上下文无关语言的层次上. 因此我们认为这个猜测很可能是成立的.

在[48]中引入广义移位算子来研究区间映射中的语言复杂性问题. 在非单值映射的情况找到了上下文无关语言的例子. 图 12-2 就是在[48]中的一个例子. 其中的映射 f 由直线

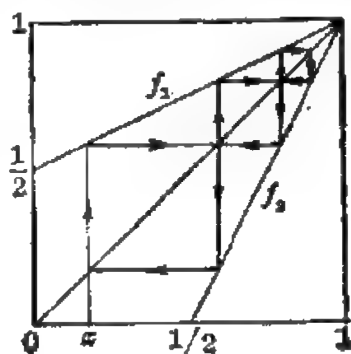


图 12-2

$$f_1(x)=(1+x)/2 \qquad 和 \qquad f_2(x)=2x-1$$

组成. 因此在区间[0.5, 1)上，f 不是单值的. 从[0, 0.5]中的某个点出发，可以得到图 12-4 中的轨. 当然，这时由于 f 非单值，轨 $f^n(x)$含有不确定性. 我们约定当轨与 f_1 所代表的线段相遇时用左括号“(”来记，而当轨与 f_2 所代表的线段相遇时则用右括号“)”来记，这样就可以将轨转化为由符号(和)组成的踪迹. 图中所示的轨对应于踪迹

$$((\ )((\ )(\ )))((\ ))((\ )(\ )(\ )).$$

由于轨的重复部分在图上看不出来，因此也可作其他理解，但括号的层次只有三层深度是这样的轨所共有的.

以上将轨转化为符号串的方法与§5.2中的做法完全一致.

如果研究所有起点和终点重合的轨所对应的符号串，则所得到的即是在$S=\{(,)\}$上的狄克语言，即在§3.2中介绍过的上下文无关语言. 但它不是正规语言.

然而，由于通常研究的区间映射均为单值映射，在这个意义上，[54]中的猜测仍有待证明或推翻.

接下来考虑的是上下文有关语言类. 从本章已经看到，大量的$\mathscr{L}$(KS)是上下文有关语言，但不是上下文无关语言. 利用在§2.6中的线性有界自动机，容易证明，凡是由同态生成的揉序列，相应的语言的复杂层次不会超过上下文有关语言类. 这方面的完整刻划是什么还很不清楚. 从§12.1可以看出，按一定的方式交替使用两个不同的同态，可以产生这个类中的大量实例.

再高一个层次是递归可枚举语言以及递归语言(见§2.4和§2.5). 容易看出，如果对于"给定一个揉序列 KS$=a_1a_2\cdots a_n\cdots$"这句话理解为给出a_n的计算方法，即可以用一个图灵机来计算每个a_n，那么语言$\mathscr{L}$(KS)的复杂性不会超出递归语言类.

但是还没有例子，不论是具体的还是抽象的，可以证明存在$\mathscr{L}$(KS)，它是递归语言或递归可枚举语言，但却不是上下文有关语言.

最后，我们引用在[47]中的结果，即证明存在$\mathscr{L}$(KS)，它的复杂性层次为非递归可枚举语言，即所谓不可计算的复杂层次.

实际上，由于乔姆斯基层次中的所有语言只有可列个，只要证明$\mathscr{L}$(KS)全体为不可列集，上述论断就成立.

关于这个不可列性的证明有几种方法. 利用在§6.4中的讨论，只要证明存在不可列个不相同的揉序列就够了. 利用二次方映射$f(x)=bx(1-x)$当b从0到4时拓扑熵为b的连续函数，且从值0单调增加地达到$\log 2$，我们就知道存在不可列个不同的揉序列. 或者我们也可以利用带参数的帐棚映射F_s: $[0, 1]\to[0, 1]$, $1\leqslant s\leqslant 2$，其中

$$F_s(x)=\begin{cases}sx, & \text{当 } 0\leqslant x\leqslant\dfrac{1}{2}\text{ 时},\\ s(1-x), & \text{当 } \dfrac{1}{2}<x\leqslant 1\text{ 时}.\end{cases}$$

利用 F_s 的拓扑熵为 $\log s$，同样推出存在不可列个不同的揉序列.

以上有关的材料可参见[21].

此外，也可以引用[68]中的结果，即从两个周期窗的同态出发，就可生成不可列个凝聚点，它们对应于不可列个揉序列.

这样我们就看到，应用以上关于不可列性的抽象讨论，已经证明在单峰映射的形式语言 $\mathscr{L}$(KS) 中存在着不可计算的复杂性. 当然，我们不可能给出具有这样复杂程度的一个具体的揉序列. 因为一旦能够给定出这样的 KS，那么如上所说，$\mathscr{L}$(KS)的复杂性层次不会超过递归语言类，而且非常可能它是上下文有关语言.

第5章

多样性与禁止字

本章研究语言复杂性的新的刻划方法.

第一种方法是研究语言中长度为 n 的字的个数 N_n 当 $n\to\infty$ 时的增长数 s. 将它取对数,就得到熵 $h=\log s$. 在§13中讨论了熵的计算方法,并且说明这样定义的熵对于区间迭代映射的动力系统即是拓扑熵. 在§14中研究熵的数值大小的动力学意义. 在本书的附录D中证明语言的生成函数 $N(t)$ 与揉行列式之间的一个重要公式.

第二种方法是研究在语言中不出现的最短符号串,即所谓禁止字. 讨论了单峰映射的形式语言中的有限补语言与无限补语言. 对于其中的正规语言,得到了禁止字的完整结构. 研究了禁止字集合的复杂性,以及它与原来语言的关系. 研究了非正规语言的禁止字的具体例子,提出了一个指标.

§13 形式语言的熵

本节从一个形式语言所含符号字的多样性引入熵的概念,介绍它的基本性质、解析理论,并指出与拓扑熵的等价性. 主要材料来自[21],但在处理方法上不全相同.

§13.1 熵的定义

从第4章已经看到,对于大量存在的非正规语言来说,缺乏有效的复杂性刻划方法. 这时没有如§9中的最小有限自动机那样的工具可以利用.

本节对于语言中所含符号串的多样性提出一种刻划方法，它来源于信息论．以下简称为**形式语言的多样性**（参看[71]～[76]）．

对于只含有限个符号串的语言，符号串的数量就是语言多样性的最直接的度量．对于含有无限多个符号串的语言，则可以统计在语言中长度恰为 n 的符号串的个数 N_n，这样就得到一个数列 $\{N_n\}_{n\geq 1}$．如果这个语言含空串 ε，则令 $N_0=1$．曼代尔勃洛特(B. Mandelbrot) 在 [72] 中称 $\{N_n\}_{n\geq 0}$ 为语言的结构函数 (Structure Function)．

考虑 $S=\{0, 1\}$上的语言 $L=S^*$，则容易看出 $N_n=2^n$，$n\geqslant 0$．显然在只由两个符号生成的所有形式语言中，这个语言在多样性方面已达到最大可能．对 S^* 中的其他语言则 $N_n\leqslant 2^n$，$n\geqslant 0$．对于由 k 个符号生成的形式语言，相应地有不等式 $N_n\leqslant k^n$ 成立．

现在采取在[21]等文献中的方法，定义数列$\{N_n\}_{n\geq 0}$ 的增长数 s．当 n 无限增长时，N_n 的性态与 s^n 类似．因此 s 的数值大小反映了 N_n 的变化，即反映了语言的多样性．

增长数的数学定义是

$$s=\lim_{n\to\infty}\sqrt[n]{N_n}=\lim_{n\to\infty}\frac{N_{n+1}}{N_n}.$$

对于数列$\{N_n\}_{n\geq 0}$，这两个极限必定存在．实际上，利用在 §7.4 中的性质 1，如果将一个长度为 $n+m$ 的符号串 z 写成

$$z=uv,$$

其中 $|u|=n$，$|v|=m$，那么就有

$$z\in L \Rightarrow u、v\in L.$$

这就得到不等式

$$N_{n+m}\leqslant N_n\cdot N_m.$$

由此出发，按照数学分析中的标准方法，就可以证明上述两个极限存在且相等．这样我们就知道对于 $\mathscr{L}$(KS)，增长数总是有定义的．由于在下一章中讨论的形式语言也具有 §7.4 中的基本性质，

因此增长数的定义也没有问题.

利用§7.4的性质2,易见有

$$N_{n+1} \geqslant N_n \qquad (其中\ n \geqslant 0).$$

此外,从增长数定义可见,$s \geqslant 1$,因此,对于$S=\{0, 1\}$上的任何语言,增长数s满足不等式

$$1 \leqslant s \leqslant 2.$$

现在定义

$$h = \log s,$$

称它为语言L的熵. 可以将h的计算公式写为

$$h = \lim_{n \to \infty} \frac{\log N_n}{n}.$$

这里与本书其余部分一样,所用的对数均以2为底.

从关于s的不等式得到关于h的不等式

$$0 \leqslant h \leqslant 1.$$

在信息论中,称上述h为信道容量. 采用[71]的概念,我们将二中择一时达到最大程度的不确定性取为信息量单位,即一个比特(Bit).这就是说,对于可能出现的符号,有两种程度相同的可能性. 相仿地,如果有k种可能性,且可能程度相同,则其不确定性为$\log k$. 我们也将不确定性说成是所含的信息量.

现在如果将形式语言看成是从一个信源发出的符号串集合,例如用某种生成语法或其他手段不断地生成各个符号串并发送出去,而在另一端用相应的自动机来接受它们,那么上述熵$h=\log s$就是在这样的传输通道(即信道)中一个符号平均含有的信息量或不确定性. 因此,在这个意义上我们可以将h称为信道容量.

当然,这里没有考虑各个符号与符号串的实际出现概率,因此,h提供的是信道的最大可能容量. 这是由形式语言L所完全确定的.

在[41]、[45]、[77]、[78]中考虑了计入概率因素的分析方法,本书不作介绍.

§ 13.2 关于熵的一些性质

设 L_1 和 L_2 是在符号集 S 上的两个语言. 将它们的增长数和熵分别记为 s_1、h_1 和 s_2、h_2, 从定义可见, 如果有

$$L_1 \subseteq L_2 (\subseteq S^*),$$

则成立不等式

$$s_1 \leqslant s_2, \qquad h_1 \leqslant h_2.$$

现在设 L_1 和 L_2 是分别由揉序列 KS_1 和 KS_2 确定的形式语言, 即

$$L_1 = \mathscr{L}(KS_1), \qquad L_2 = \mathscr{L}(KS_2).$$

那么从语言的定义(见§ 6.1)直接看出, 如果 $KS_1 \leqslant KS_2$, 则 $L_1 \subseteq L_2$. 因此, 增长数和熵都是揉序列的单调增加函数(严格地说是单调不降函数).

在[21]中证明, 二次方映射 $f(x) = bx(1-x)$ 的揉序列关于参数 b 是单调增加的, 因此, 如果将相应的形式语言的增长数和熵记为 b 的函数 $s(b)$ 和 $h(b)$, 则它们都是 b 的单调增加函数. 已经证明, 它们关于 b 都是连续的. 在图 13-1 中作出了函数 $h(b)$ 在 $3.2 \leqslant b \leqslant 4$ 时的图象. 它是一条连续曲线, 在 $b < b_\infty (\approx 3.57)$ 时与 b 轴重合, 在每个周期窗口处为常数. 由于这些周期窗的宽度大多数都很小, 我们只在图上标出了周期 3 窗口对应的水平直线段. 这时, 参数 b 的范围大致是 3.828…到 3.857…, $h = 0.694$…. 关于这一点的证明, 见§ 14.4.

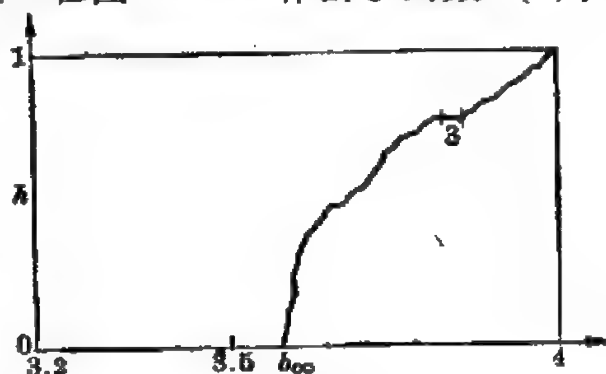


图 13-1

将图 13-1 所展示的多样性刻划与第 3、4 章中的语言复杂性分析作比较, 可见二者是很不平行的. 在乔姆斯基层次中的复杂性是一个语言用语法生成时的困难程度的反映, 它也可以从接受(或识别)语言的自动机的复杂程度得到反映. 这与语言中所含符

号字的多少没有直接关系.

具体来说，上一章中详细研究过的费根鲍姆吸引子一般都认为是一个最为简单的准周期系统. 它所包含的周期轨只有以$t_n^{\infty}(n\geqslant0)$为踪迹的那些轨，即对每个2的幂$2^n$只存在一种周期轨，再加上其他不动点. 对于二次方映射，从图10-1可以看到，费根鲍姆点b_∞处在进入混沌的门槛上. 在$b>b_\infty$时出现了与$b<b_\infty$时完全不同的复杂行为. 在这一点上与图13-1是一致的. 在下面将严格证明，语言$\mathscr{L}(t_\infty)$的熵为零. 从熵的信息论意义来看，这表明在极限的意义上，每一个符号所含的平均信息量为零. 实际上，可以认为这暗示了语言$\mathscr{L}(t_\infty)$的结构并不太复杂. 这也是在§10中很容易将语言$\mathscr{L}(t_\infty)$的结构分析清楚的一个根本原因. 但是从§11的分析已经知道，语言$\mathscr{L}(t_\infty)$在乔姆斯基层次体系中已属于相当复杂的上下文有关语言的范畴.

另一方面，从图13-1则可知在第3章中所讨论的所有正规语言，除了$\mathscr{L}(t_n^{\infty})(n\geqslant0)$之外，它们的熵都是正的. 特别是在KS$=10^{\infty}$时，熵达到最大值. 从第3章的分析可以看出，尽管某些语言中所含的符号串在数量上要比$\mathscr{L}(t_\infty)$的情况多得多，但语言的生成和接受都属于乔姆斯基层次中的最低层.

各种复杂性刻划手段之间的不平行性，是在复杂性研究中屡见不鲜的现象，甚至可以说，是必然现象. 反之，如果存在一种统一方法可以将某一类系统(或问题)完全刻划出来，甚至于可以按某个数量特征将研究对象从复杂程度低到高线性排列起来，那么，这类系统(或问题)应当说不能算作很复杂了. 我们已经看到，对于区间映射，甚至是一个二次方映射族，情况完全不是如此(参见§7.1中的讨论)，各种方法刻划了系统的不同侧面，相互之间往往是很不平行的.

§13.3 计算熵的几个例子

先引进一个新的数列$\{S_n\}_{n\geqslant1}$，它在下面很有用处.

S_n是语言 L 中长度为 n 且从符号 1 开始的符号串的个数. 显然, 有 $S_n \leqslant N_n$ 成立. 对于语言 $\mathrm{L}=\mathscr{L}(\mathrm{KS})$, 存在更为确定的关系:

$$S_n = N_n - N_{n-1},\ n \geqslant 1.$$

实际上, 如果 $z \in \mathrm{L}$, $|z|=n$, 且从符号 0 开始, 则可以记 $z=0z'$. 这时 $z' \in \mathrm{L}$, $|z'|=n-1$. 另一方面, 从 $z' \in \mathrm{L}$ 可以推出 $0z' \in \mathrm{L}$, 这只要利用在§7.5 中的判定法则就可以得到. 由此可见, 在 N_n 个长度为 n 的字中间, 从符号 0 开始恰有 N_{n-1} 个, 因此得到

$$S_n = N_n - N_{n-1}.$$

可以证明, 数列 $\{S_n\}_{n \geqslant 1}$ 的增长数与$\{N_n\}_{n \geqslant 0}$ 的增长数相同. 因此, 可以用$\{S_n\}_{n \geqslant 1}$ 计算增长数 s 和熵 h.

[例 1] 计算语言 $\mathrm{L}=\mathscr{L}((10)^{\infty})$的熵.

接受这个语言的最小有限自动机见图 9–2的$|x|=4$ 的第一个例子. 容易写出 L 的正规表达式为

$$0^*+0^*11^*+0^*11^*0+0^*11^*0(10)^*+0^*11^*0(10)^*1$$

(参见§1.5与[1]).

但在计算 L 的熵时, 并不需要这些知识.

直接计算 S_n, $n \geqslant 1$. 考虑长度 $n \geqslant 2$ 且以符号串 11 开始的字的个数. 如 $z \in \mathrm{L}$, $z=1z'$, 且 z' 也从 1 开始, 那么 $z' \in \mathrm{L}$. 利用§7.5, 可见反之也对. 因此在 L 中这样的字有 S_{n-1} 个. 由此可见, 在 L 中从符号串 10 开始的字恰有 $S_n - S_{n-1}$ 个. 由 $\mathrm{KS}=(10)^{\infty}$, 从 10 开始的允许字只可能是揉序列本身(参见附录 C.4的例 2). 这样就建立起递推关系

$$S_n - S_{n-1} = 1, \qquad n > 1.$$

从 $S_1=1$ 可见它的解为 $S_n=n+1$. 这样就得到增长数与熵为

$$s=\lim_{n\to\infty}\frac{S_{n+1}}{S_n}=1 \qquad h=\log 1=0.$$

或者直接从

$$h=\lim_{n\to\infty}\frac{\log S_n}{n}$$

求出 h 的值为 0.

另一个方法是建立关于$\{N_n\}_{n\geq 0}$ 的二阶递推关系

$$N_n-2N_{n-1}+N_{n-2}=1, \qquad n\geqslant 2.$$

它也可以从上述关于 S_n 的递推式导出. 这是一个二阶差分方程. 令 $N_n=\alpha^n$ 代入, 得到特征方程为

$$\alpha^2-2\alpha+1=0,$$

它以 $\alpha=1$ 为二重根. 这样就可知增长数 $s=1$ 和熵 $h=0$. 从 $N_0=1$ 和 $N_1=2$ 出发, 可以解出

$$N_n=\frac{1}{2}(n^2+n+2),$$

即是一个二次多项式, 因此 $\{N_n\}_{n\geq 0}$ 当 $n\to\infty$ 时的增长速度比指数规律要慢得多.

对于二次方映射 $f(x)=bx(1-x)$, $\mathrm{KS}=(10)^\infty$ 对应于参数 b 的范围为$(1+\sqrt{5}, 3.2428\cdots)$.

使用 § 10.2 中的记号, 上述揉序列即为 t_1^∞. 同样可以证明, 对一切 t_k, $k\geqslant 0$, 语言 $\mathscr{L}(t_k^\infty)$的增长数都是 1, 熵都是 0. 这时$\{N_n\}_{n\geq 0}$ 的表达式都是多项式.

利用对 $k\geqslant 0$ 成立

$$t_k^\infty<t_\infty,$$

其中 t_∞ 是费根鲍姆吸引子的揉序列(见 § 10.2), 按照 § 13.2 一开始指出的单调性质, 我们不必逐个计算 $\mathscr{L}(t_k^\infty)$ 的熵, 只要证明 $\mathscr{L}(t_\infty)$的熵为 0 就够了.

[例 2] 计算费根鲍姆吸引子的熵.

从 $\mathrm{KS}=t_\infty$ 出发, 先计算在语言 $\mathrm{L}=\mathscr{L}(t_\infty)$中字长为 $2n$ 且从 10 开始的字的个数. 与例 1 中的推导一样, 可求出这个数为

$$S_{2n}-S_{2n-1}.$$

然后, 利用 § 10.2 中引入的同态

$$h=\{0\to 11,\ 1\to 10\},$$

可见当 $z\in\mathrm{L}$ 时, 也有 $h(z)\in\mathrm{L}$. 反之, 如 $z\in\mathrm{L}$, $|z|=2n$, 且从 10

开始，则也有 $h^{-1}(z)\in L$ 成立．这里的 $h^{-1}(z)$ 是长度为 n 且从符号 1 开始的字，它由 z 唯一确定．这一点从 § 10.4 的结论来看是显然的．但也可以不必利用那里关于语言 $\mathscr{L}(t_\infty)$ 的结论，直接证明当 $z\in L$ 且从 10 开始时，z 的每个奇数位必是 1．例如，用数学归纳法就很容易证明这一点．由此即可知 z 的逆同态象 $h^{-1}(z)$ 有定义．

这样就建立了关系式

$$S_{2n}-S_{2n-1}=S_n,\qquad n\geqslant 1.$$

这不是在通常意义下的差分方程．我们不来求它的解，而只证明

$$\lim_{n\to\infty}\frac{S_{n+1}}{S_n}=1$$

成立，从而熵 $h=0$．

用反证法．如果上述极限记为 s，且 $s>1$，则可以写出

$$1-\frac{S_{2n-1}}{S_{2n}}=\left(\frac{S_n}{S_{n+1}}\right)\cdot\left(\frac{S_{n+1}}{S_{n+2}}\right)\cdots\left(\frac{S_{2n-1}}{S_{2n}}\right).$$

当 $n\to\infty$ 时，左方的极限为 $1-1/s>0$．对于右方，先取 N，使当 $n>N$ 时，右方的每一个因子都小于 $2/(1+s)<1$．这时在 $n>N$ 时，右方就小于

$$(2/(1+s))^n.$$

由于 $s>1$，当 $n\to\infty$ 时它的极限为 0．引出矛盾．

关于以上两个例子的讨论，方法与[79]类似．此外，关于熵的计算，可参看[80]．

§ 13.4　伴随矩阵方法

对于正规语言，可以利用接受它的最小有限自动机来计算熵．这时的语言不限于单峰映射所生成的语言，但必须是正规语言．

以 $L=\mathscr{L}((10)^\infty)$ 为例．从接受 L 的有限自动机（见图 9-2）出发，写出它的伴随矩阵

$$X=\begin{pmatrix}1&1&0&0\\0&1&1&0\\0&0&0&1\\0&0&1&0\end{pmatrix}.$$

这里的规则是：将自动机的所有与终止状态对应的结点编号，如果第 i 个结点有通往第 j 个结点的弧时，矩阵 X 的 (i, j) 元为 1，否则为 0. 设自动机的初态为第一个结点.

从矩阵的乘法规则可见，幂 X^n 的第一行元素代表了从初态出发到每一个终止状态的路径数目，这里路径长度为 n. 由于所用的自动机是接受语言 L 的最小确定性有限自动机，将 X^n 的第一行元素相加，就得到 N_n.

可以证明，当 $n\to\infty$ 时，X^n 的第一行元素之和与 X 的最大正特征值的 n 次幂是等价无穷大. 因此，这个特征值就是 $\{N_n\}_{n\geq 0}$ 的增长数 s.

因此只要计算伴随矩阵的最大特征值，再取对数，就求出了熵.

以上述 X 为例，求出特征方程为

$$\det(\lambda I - X) = (\lambda-1)^3(\lambda+1),$$

因此 $s=1$, $h=0$.

在动力系统中，把上述伴随矩阵叫做斯蒂芬矩阵（Stefan Matrix）. 对此，可参见[68]、[26]. 计算类似问题的一般性讨论见[81]. 关于非正规语言的熵的计算有很大困难，可参见[75]、[76]. 但这是就一般形式语言而说的. 关于单峰映射的形式语言，熵的计算方法将在下面介绍.

§13.5 生成函数与揉行列式

由数列 $\{N_n\}_{n\geq 0}$ 可以定义一个函数

$$N(t)=\sum_{n=0}^{\infty}N_n t^n=1+N_1t+N_2t^2+\cdots,$$

称为语言 L 的结构生成函数，简称为生成函数．利用极限

$$\lim_{n\to\infty}\frac{N_{n+1}}{N_n}=\lim_{n\to\infty}\sqrt[n]{N_n}=s,$$

可见 $N(t)$ 是一个收敛半径为 s^{-1} 的解析函数．

对于语言 $L=\mathscr{L}(KS)$，可以从揉序列 KS 出发直接计算生成函数 $N(t)$．

按照[21]，引入揉行列式(Kneading Determinant)

$$D(t)=1+\varepsilon_1 t+\varepsilon_1\varepsilon_2 t^2+\cdots+\varepsilon_1\varepsilon_2\cdots\varepsilon_n t^n+\cdots,$$

其中 $\varepsilon_i=\varepsilon(a_i)(i\geqslant 1)$，$a_i$ 是揉序列(它不含 c)

$$KS=a_1a_2\cdots a_i\cdots$$

中的第 i 个符号，函数 ε 的定义为

$$\varepsilon(1)=-1,\qquad \varepsilon(0)=+1,$$

它反映单峰映射的左枝和右枝的不同单调性质．在[21]中对于区间上的一般多峰映射建立了符号动力学的基本理论，其中的 $D(t)$ 是从一个行列式得到的，因此称之为揉行列式．我们沿用这个名称．

从揉序列 $KS=a_1a_2\cdots a_i\cdots$ 出发，求出 $\{\varepsilon_i\}_{i\geqslant 1}$ 与 $\prod_{i=1}^{n}\varepsilon_i(n\geqslant 1)$，就可以得到 $D(t)$．

可以建立语言 $L=\mathscr{L}(KS)$ 的生成函数 $N(t)$ 与揉行列式 $D(t)$ 之间的关系：

$$N(t)=\frac{1}{2(1-t)}+\frac{1}{2(1-t)D(t)}.$$

我们将这个公式的证明写在附录 D 中．这里只对于这个公式作解释，在 § 14 将举出大量例子．

将 t 看成复变量．由于 $D(t)$ 的系数都是 ± 1，因此 $D(t)$ 是收敛半径不小于 1 的解析函数．另一方面，已知 $N(t)$ 的收敛半径为 $s^{-1}(\leqslant 1)$．因此上述公式在 $|t|<s^{-1}$ 时成立．

由于生成函数 $N(t)$ 的展开式中每个系数为正值，因此由解析函数理论知道 $t=s^{-1}$ 恰好是 $N(t)$ 的一个极点．从上述公式可

见，$D(t)$ 在 $|t|<s^{-1}$ 时没有零点．同时，如 $s>1$，则 $t=s^{-1}$ 必是 $D(t)$ 的一个零点．

总结以上讨论，对于语言 $L=\mathscr{L}(KS)$，可以从揉序列 KS 直接写出 $D(t)$．然后计算 $D(t)$ 在区间[0, 1]上的最小零点，将它取倒数，就得到语言的增长数 s，同时得到熵 $h=\log s$．在[80]中有关于这类计算的大量例子．在[82]中对误差作了分析．

§13.6 与拓扑熵的等价性

在这一小节中我们将介绍动力系统的拓扑熵，并且指出，对于语言 $L=\mathscr{L}(KS)$，在本节一开始所定义的熵与拓扑熵是等价的．对于区间上的一般多峰映射，这仍然成立，可参看[83]～[85]．

设映射 T 将集合 X 映入自身，将由迭代生成的动力系统记为 (X, T)，将从点 $x\in X$ 出发的轨记为

$$T^{*}(x)=\{x, T(x), \cdots, T^{n}(x), \cdots\},$$

动力系统的拓扑熵是关于轨 $T^{*}(x)$ 的不确定性（或所含信息量）的一个度量．它的严格定义见原文[85]．以下只是对于拓扑熵的一个通俗解释．

利用符号动力学方法，将 X 作某种有限划分，并将这个划分记为 α．设 α 将 X 划分成 $N(\alpha)$ 个子集，并用 $N(\alpha)$ 个不同符号来标记这些子集，那么就可以与§5.2中一样，将轨转化成符号序列．但在这里我们不这样做下去，而是将划分 α 看成是对 X 中的点的一个测量过程，并关心所能得到的信息量．

对于点 x，由于我们的测量方法只能确定到 x 属于 $N(\alpha)$ 个子集中的哪一个，因此按照信息论，所含的信息量可记为

$$H(\alpha)=\log N(\alpha).$$

这样的测量可能是粗糙的．现在接连测量从 x 出发的轨上的前 n 个点，即

$$x, Tx, \cdots, T^{n-1}x,$$

就可以得到长度为 n 的一个符号序列．我们将要说明，利用这 n

次测量的结果，我们关于 x（或者轨 $T^n(x)$）的不确定性就要小得多了.

引进一点数学记号是必要的. 记 $T^{-i}(\alpha)$ 是划分 α 在映射 T 下的第 i 次原象，它也是关于集合 X 的一个划分. 又引入记号

$$\bigvee_{i=0}^{n-1} T^{-i}(\alpha)$$

表示将所有划分 α、$T^{-1}(\alpha)$、…、$T^{-n+1}(\alpha)$ 合并在一起的一个划分. 设以上这些划分均为有限划分. 对轨 $T^n(x)$ 的前 n 个点作测量，得到长度为 n 的符号序列，它与上述划分的某一个子集相对应.

可以看出，当 n 无限增大时，划分 $\bigvee_{i=0}^{n-1} T^{-i}(\alpha)$ 越来越细，这反映出我们得到越来越多的信息. 在很多场合，作为极限情况，有可能完全确定初始点 x，或者说轨 $T^n(x)$ 本身.

反映动力系统 (X, T) 行为的一个方法，即是计算在上述测量过程中的信息量. 采用与上面相同的记号 N 和 H，则在 n 次测量中的平均信息量（或不确定性）为

$$\frac{1}{n} H\left(\bigvee_{i=0}^{n-1} T^{-i}(\alpha)\right) = \frac{1}{n} \log N\left(\bigvee_{i=0}^{n-1} T^{-i}(\alpha)\right).$$

令 $n \to \infty$，就得到只与划分 α 有关的平均信息量的极限，记为

$$h(T, \alpha) = \lim_{n\to\infty}\left(\frac{1}{n} H\left(\bigvee_{i=0}^{n-1} T^{-i}(\alpha)\right)\right).$$

但是，由 α 所代表的测量方法未必是最好的，因此，我们再引进

$$h(T) = \sup_{\alpha} h(T, \alpha).$$

这就是动力系统 (X, T) 的拓扑熵. 它的严格定义是在[85]中首先提出来的.

由此可见，拓扑熵 $h(T)$ 反映了动力系统 (X, T) 的动力学行为中的平均信息量，或者说平均不确定性，这也就是对轨道的多样性的一种可能度量.

如果在集合 X 中引进距离，对于子集合的大小引进直径的概念，然后定义划分 α 的直径是它所含的子集合的直径的最大值，

那么在适当条件下可以证明，如果当 $n\to\infty$ 时，上述加细划分 $\bigvee_{i=0}^{n-1} T^{-i}(\alpha)$ 的直径趋于 0，就有

$$h(T)=h(T, \alpha).$$

实际上这是拓扑熵的主要计算方法. 它具有明显的动力学意义. 实际上，如果对轨 $T^{*}(x)$ 的连续观测在极限意义上能完全精确地确定 x(或 $T^{*}(x)$)的话，则这样的测量方法(或者说划分 α)就提供了最大可能的信息量.

从以上介绍可见，毫不奇怪的是，关于单峰射的语言 $\mathscr{L}$(KS) 在 § 13.1 中定义的熵与这里所介绍的拓扑熵恰好相等. 这个事实的证明见[83]、[84]. 关于拓扑熵还可以参看[86]、[87].

在[21]中对于区间上的分段单调连续映射 f 定义 $l(f)$，它是区间的一个划分 α 所对应的子区间个数 $N(\alpha)$，要求在每个子区间上 f 单调，并且使划分个数达到极小. 然后定义数列 $\{l(f^n)\}_{n\geq 1}$ 的增长数 s 和拓扑熵 $h=\log s$. 这与本节的做法几乎相同. 对于单峰映射 f，有 $l(f)=2$. 容易建立公式

$$l(f^n)=2N_{n-1},$$

并建立类似的其他联系.

在动力系统研究中经常引入拓扑共轭的概念. 可以证明拓扑熵是拓扑共轭下的不变量.

由此可见，由 $\mathscr{L}$(KS)引入的熵也是拓扑共轭下的不变量. 但实际上这个熵仅由揉序列确定. 因此当两个单峰映射具有相同揉序列，但并非拓扑共轭时，它们的熵仍然相等. 这里可参看 § 5.4 中的图 5-1 和 § 7.1 中的图 7-1 所提供的例子. 从这个角度看来，直接从语言 $\mathscr{L}$(KS) 定义熵 h 是非常自然的.

如果在作划分 α 时，同时对 α 的各个子集赋以不同的权重，就可以自然地导出测度熵. 在历史上它的出现早于拓扑熵(参见[87]). 已经证明，拓扑熵是所有可能的测度熵中最大的一个. 因此对熵 h 的计算给出了测度熵的一个上界估计. 又由于对于单峰

映射来说，测度熵与李雅普诺夫指数一般相等，因此熵 h 也是对于李雅普诺夫指数的一个上界估计．这方面可参看[88]．

§14 熵的计算和意义

本节以单峰映射的许多结果说明，熵不仅是对于语言中字的多样性或动力系统中轨道的多样性的刻划，同时还直接反映了动力系统的许多重要性质．本节材料主要取自[21]．

§14.1 费根鲍姆吸引子的熵

应用揉行列式 $D(t)$，我们对于费根鲍姆吸引子的熵给出新的计算方法．读者可以将它与§13.3中的例2作比较．

从揉序列

$$t_\infty=10111010\cdots$$

可以直接写出揉行列式

$$D(t)=1-t-t^2+t^3-t^4+t^5+t^6-t^7-\cdots.$$

利用在§10.3中指出的序列 t_∞ 与图厄-莫尔斯序列的关系，可以看出 $D(t)$ 的系数恰好是由符号 $+1$ 和 -1 生成的图厄-莫尔斯序列．这样就可以得到 $D(t)$ 的无穷乘积形式：

$$D(t)=(1-t)(1-t^2)(1-t^4)\cdots(1-t^{2^n})\cdots.$$

应用微积分中的判别法则，可知当 $0\leqslant t<1$ 时这个无穷乘积收敛．我们知道，按照无穷乘积的收敛定义，这已表明 $D(t)$ 在 $0\leqslant t<1$ 时没有零点．根据在§13.5中的分析，我们得到 $\mathscr{L}(t_\infty)$ 的增长数 $s=1$，熵 $h=0$．

在[21]中给出了 $D(t)$ 的上述无穷乘积形式．在[89]中指出了它与图厄-莫尔斯序列的联系．

可以指出，对于语言 $\mathscr{L}(t_\infty)$ 来说，数列 $\{N_n\}_{n>0}$ 的增长速度超过任何多项式，但低于指数函数．

§14.2 关于熵的两个计算公式

现在研究当 $\mathscr{L}$(KS)为正规语言时的熵的计算公式. 由于任何揉序列可以用周期揉序列来逼近, 根据[82]的研究, 可以对任何揉序列 KS 近似地计算出语言 $\mathscr{L}$(KS)的熵.

利用 §8 中的定理 1, 在 $\mathscr{L}$(KS)为正规语言时, KS 不是周期序列就是终极周期序列. 统一这两种情况, 可将揉序列写成

$$\mathrm{KS}=a_1\cdots a_m(a_{m+1}\cdots a_n)^\infty.$$

在 $m=0$ 时这就是周期序列. 不妨取 $a_{m+1}\cdots a_n$ 为极小串, 即长度极小的偶串(参见 §9.2). 在终极周期而非周期序列的情况, 我们要求 $a_m\neq a_n$, 这时 KS 的写法是唯一确定的.

在 $m=0$ 时, 按以上条件可以直接得到

$$D(t)=(1+\varepsilon_1 t+\cdots+\varepsilon_1\cdots\varepsilon_{n-1}t^{n-1})/(1-t^n).$$

因此只要计算多项式

$$p_1(t)=1+\varepsilon_1 t+\cdots+\varepsilon_1\cdots\varepsilon_{n-1}t^{n-1}$$

在[0, 1]中的最小正零点.

在 $m>0$ 时, 同样可以得到

$$D(t)=1+\varepsilon_1 t+\cdots+\varepsilon_1\cdots\varepsilon_{m-1}t^{m-1}+\varepsilon_1\cdots\varepsilon_m t^m(1+\varepsilon_{m+1}t+\cdots+\varepsilon_{m+1}\cdots\varepsilon_n t^{n-m-1})/(1-t^{n-m}).$$

因此只要计算多项式

$$p_2(t)=1+\varepsilon_1 t+\cdots+\varepsilon_1\cdots\varepsilon_{n-1}t^{n-1}-t^{n-m}(1+\varepsilon_1 t+\cdots+\varepsilon_1\cdots\varepsilon_{m-1}t^{m-1})$$

在[0, 1]中的最小正零点.

将所得的零点记为 s^{-1}, 则 s 即为增长数, 而 $h=\log s$ 即是语言 $\mathscr{L}$(KS)的熵.

举两个最简单的例子.

对于 KS=(10)$^\infty$, 即 §13.3 的例 1. 有 $m=0$, $n=4$. 计算出 $\varepsilon_1=-1$, $\varepsilon_2=+1$, $\varepsilon_3=-1$, 就得到

$$p_1(t)=1-t-t^2+t^3=(1-t)(1-t^2).$$

可见$s=1$, $h=0$.

对于 $KS=10^\infty$, 则 $m=1$, $n=2$, 可计算出

$$p_2(t)=1-2t.$$

因此$s=2$, $h=1$. 这即是在§13.1中所说的熵达到极大的情况.

在[26]、[68]中指出, 对于语言$\mathscr{L}(KS)$可以从揉序列 KS 出发, 直接写出斯蒂芬矩阵, 而不必如§13.4那样利用最小有限自动机. 然后只要计算矩阵的最大正特征值就可以求出增长数 s. 这些结果与上面所得的多项式$p_1(t)$和$p_2(t)$完全一致, 只要作$\lambda=1/t$的代换即可.

§14.3 熵与奇周期轨

我们要证明, 对于语言$\mathscr{L}(KS)$来说, 熵的数值大小与奇周期允许字的存在有密切关系. 利用§6.3的定理2, 即可以判定原来的单峰映射是否存在奇周期轨.

先证明一个命题作为准备工作.

命题 设 $KS=(101^{n-2})^\infty$, 其中 n 是大于2的奇数, 那么语言$\mathscr{L}(KS)$的熵$h=\log\omega$, 这时的ω是方程

$$\lambda^n-2\lambda^{n-2}-1=0$$

的(唯一)正根.

这个结果最早见于[52].

利用§14.1中的公式, 直接写出

$$p_1(t)=1-t-t^2+t^3-t^4+\cdots-t^{n-1}.$$

乘以$1+t$, 就有

$$(1+t)p_1(t)=1-2t^2-t^n.$$

再令$t=1/\lambda$, 可见增长数s是方程$\lambda^n-2\lambda^{n-2}-1=0$的正根. 容易验证这个方程只有一个正根.

固定奇数$n(>2)$. 在所有最小周期为n的周期允许字中, $(101^{n-2})^\infty$具有特殊意义. 实际上. 如对于某个揉序列 KS, 存在最小周期为n的允许字x^∞, $|x|=n$, 那么在$x=M(x)$时, 就有

$$101^{n-2} < x$$

成立. 这时$(101^{n-2})^\infty$也是允许字.

证明是容易的. 这时的串x必从10开始, 不妨记为

$$x = 101^{m_1}01^{m_2}\cdots 10^{m_{k-1}}01^{m_k},$$

其中$m_i \geq 0$, $i=1, \cdots, k$. 用反证法. 如果$x < 101^{n-2}$, 则m_1为奇数. 又利用$x = M(x)$, 即x为循环移位最大字(参见附录B.5), 可看出m_2、$\cdots$、m_{k-1}也都只能是奇数. 又从

$$101^{m_k+1}\cdots < 101^{m_1}\cdots,$$

可看出m_k是偶数. 但这样就导致长度$|x| = n$不会是奇数, 引出矛盾.

如果将上述x^∞和$(101^{n-2})^\infty$看成为揉序列, 则这意味着在二次方映射$f(x) = bx(1-x)$的分岔图10-1(或12-1)上, 在所有奇周期$n(>2)$的周期窗中, 与$(101^{n-2})^\infty$对应的窗的位置在最左方, 即对应的参数值b最小. 对于$n=3$, 由于周期三窗只有一个, 这个结论是平凡的.

现在来介绍本小节的主要结果, 即证明单峰映射存在奇周期轨的充分必要条件是熵

$$h > \log\sqrt{2}.$$

这里的奇周期$n>2$, 即除了不动点之外.

必要性　记揉序列为KS. 如果存在奇周期n的轨, $n>2$, 则从§6.3的定理2, 存在一个允许字x^∞, $|x| = n$, 且有$x = M(x)$. 利用上述讨论, 有

$$(101^{n-2})^\infty \leq x^\infty \leq \mathrm{KS}.$$

利用$(101^{n-2})^\infty$本身也可作为揉序列看待, 同时利用在§13.2一开始所说的熵的单调性, 就推出

$$h \geq \log\omega,$$

其中ω是方程$\lambda^n - 2\lambda^{n-2} - 1 = 0$的(唯一)正根. 用$\lambda = \sqrt{2}$代入方程左方得到$-1$, 这就证明了$\omega > \sqrt{2}$.

充分性　从$s > \sqrt{2}$已可推出揉序列一定从符号1开始(即

KS$\neq 0^{\infty}$)。

用反证法．如果不存在任何奇周期轨，那么从§6.3的定理2，这时也不会存在任何奇周期允许字．我们证明，这将导致揉序列KS的每个奇数位为1．否则，设KS$=a_1a_2\cdots a_i\cdots$，且将第一个为0的奇数位记为a_{2n+1}，则有

$$\mathrm{KS}=101^{n_1}0\cdots01^{n_k}a_{2n}0\cdots,$$

其中n_1、$\cdots$、n_k全为奇数．这里要指出，$n_1>1$成立．否则KS从100开始，那么$(101)^{\infty}$即是一个奇周期允许字．从$100>\mathrm{KS}$又推出$a_{2n}=1$．现在考察k．如$k>1$，则从KS为移位最大字将导致与$101^{n_k+1}0>101^{n_1}0$矛盾(因n_k为奇数)．但如果$k=1$，则KS$=101^{n_1+1}0\cdots$，n_1奇，因此存在奇周期允许字$(101^{n_1+2})^{\infty}$，又引出矛盾．

现在从KS的每个奇数位为1出发，可见

$$\begin{aligned}D(t)&=1-t\pm t^2\mp t^3\pm t^4\mp t^5\pm\cdots\\&=(1-t)(1\pm t^2\pm t^4\pm\cdots).\end{aligned}$$

由此直接看出当$0\leqslant t\leqslant 1/\sqrt{2}$时$D(t)>0$．因此增长数$s\leqslant\sqrt{2}$，即$h\leqslant\log\sqrt{2}$．这与条件$h>\log\sqrt{2}$相矛盾．

这个结论清楚地表明，熵的数值大小与动力系统的动力学行为有密切联系．在二次方映射的分岔图10-1(和图12-1)上第一倒分叉点b_{M_0}(即混沌带二合为一的点)对应的熵即是$\log\sqrt{2}$．实际上，这时的揉序列KS$=101^{\infty}$．利用§14.2中的公式，写出

$$p_2(t)=1-t-2t^2+2t^3=(1-t)(1-2t^2),$$

就可以得到$h=\log\sqrt{2}$．

从上述结论可以知道，对于二次方映射来说，在$b<b_{M_0}$时没有任何奇周期轨，而在$b>b_{M_0}$时一定有奇周期轨出现．当然，有很多方法可以得到这些结果．在这里强调的是，从揉序列出发的独立证明与具体映射无关．

§ 14.4 周期窗口的熵

现在证明，在二次方映射的分岔图上的每一个周期窗中，熵的值是不变的常数.

为此先做些准备工作.

如果语言 $\mathscr{L}(\mathrm{KS})$的熵为 h, 与倍周期分岔的同态映射为

$$g=\{0\to 11,\quad 1\to 10\},$$

则语言 $\mathscr{L}(g(\mathrm{KS}))$的熵大于等于 $h/2$. 这里的同态在§10 中已多次使用. 为了不与本章的熵的记号发生混淆，我们用记号 g 来表示它.

容易证明，$\mathscr{L}(\mathrm{KS})$中的每个字在 g 映射下的象即是语言 $\mathscr{L}(g(\mathrm{KS}))$中的字, 长度加倍. 因此，如果记 N_n 是语言 $\mathscr{L}(\mathrm{KS})$中长为 n 的字的个数，而记 N'_n 为语言 $\mathscr{L}(g(\mathrm{KS}))$中长为 n 的字的个数, 那么就有

$$N'_{2n}\geqslant N_n$$

成立. 写出

$$\frac{1}{2n}\log N'_{2n}\geqslant\frac{1}{2}\left(\frac{1}{n}\log N_n\right),$$

再令 $n\to\infty$, 就得到所要的结论.

推广上述结论. 如果在某个揉序列KS时存在周期为 $n=2^m p$ 的轨, p 是大于 2 的奇数, 那么语言 $\mathscr{L}(\mathrm{KS})$的熵满足不等式

$$h\geqslant\frac{1}{2^m}\log\omega,$$

其中 ω 是方程 $\lambda^p-2\lambda^{p-2}-1=0$ 的正根. 这里利用了§14.3 中的结论. 证明从略.

这时增长数

$$s\geqslant\omega^{\frac{1}{2^m}}.$$

由于已知 $\omega>\sqrt{2}$ 成立, 因此得到估计式

$$s\geqslant 2^{\frac{1}{2^{m+1}}}>2^{\frac{1}{2^m p}}=2^{\frac{1}{n}}.$$

现在讨论二次方映射的周期窗．设某个周期窗左端的揉序列为

$$\mathrm{KS}_1 = x^{\infty},$$

其中 x 是偶既约的循环移位最大字．如在 § 12.1 所说的那样，由于周期窗是从切分岔生成的，因此 x 总是偶串．记 $|x|=n$．

与 § 9.5(或 § 12.1)相仿地，定义同态

$$g=\{0\to x,\quad 1\to \hat{x}\}.$$

那么，周期窗右端的揉序列为

$$\mathrm{KS}_2 = g(10^{\infty}) = \hat{x}x^{\infty}.$$

它与激变相对应.

分别对 KS_1 和 KS_2 应用 § 14.2 中的公式，得到

$$p_1(t)=1+\varepsilon_1 t+\cdots+\varepsilon_1\cdots\varepsilon_{n-1}t^{n-1},$$

$$p_2(t)=(1+\varepsilon_1 t+\cdots+\varepsilon_1\cdots\varepsilon_{n-1}t^{n-1})(1-2t^n).$$

我们已知 $p_1(t)$ 的最小正零点为 s^{-1}．由于上面已经证明了 $s>2^{\frac{1}{n}}$，即 $s^{-1}<2^{-\frac{1}{n}}$，因此 $p_2(t)$ 的最小正零点也是 s^{-1}．因此就知道语言 $\mathscr{L}(\mathrm{KS}_1)$ 与 $\mathscr{L}(\mathrm{KS}_2)$ 的熵相等．利用在[21]中已经建立的结果，即熵为 b 的单调非降函数，我们就证明了在整个周期窗上熵为常数.

这个结论的反面也成立，其证明可参看[89].

不出现周期窗的一个典型例子是帐棚映射

$$F_s(x)=\begin{cases} sx, & \text{当 } 0\leqslant x\leqslant 0.5 \text{ 时;}\\ s(1-x), & \text{当 } 0.5<x\leqslant 1 \text{ 时;}\end{cases}$$

其中参数 s 的范围从 0 到 2．图 14-1 是它的分岔图．在[21] 中证明，每一个增长数 $s>1$ 的单峰映射拓扑半共轭于 F_s，这里的参数值 s 也是 F_s 的增长数．在 $0\leqslant s<1$ 时不动点 $x=0$ 是稳定的．在 $s=1$ 时区间[0, 0.5]中每个点都是不动点．关于 F_s 及其分岔图的讨

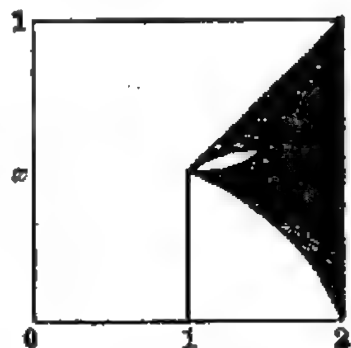


图 14-1

论，可参看[90]、[91]．可以证明，在 $s>1$ 时 F_s 的李雅普诺夫指数＝测度熵＝拓扑熵＝$\log s$．这里所用的测度是唯一存在的绝对连续遍历测度．

§14.5 熵为零的动力学意义

现在证明，单峰映射的熵为零的充分必要条件是只存在周期为 2 的幂的周期轨．

必要性实际上已经在上面的讨论中得到．如果存在周期轨，其最小周期 n 不是 2 的幂，则可写出 $n=2^m p$，p 为大于 2 的奇数．这时已知熵

$$h\geqslant\frac{1}{2^m}\log\omega,$$

ω 为方程 $\lambda^p-2\lambda^{p-1}-1=0$ 的正根．由于 $\omega>\sqrt{2}$，因此 $h>0$，引出矛盾．

充分性实际上隐含在 §10 的研究中，我们现在用数学归纳法证明．如果存在周期 2^n 的轨，但同时不存在周期不是 2 的幂的任何周期轨，那么揉序列必以 t_n 为前缀．这里的记号 t_n 在 §10.2 中已定义．

对于 $n=1$，周期 2 的允许字只有 $(10)^\infty$ 或 $(01)^\infty$，因此揉序列 KS 以 $t_1=10$ 开始是明显的．

现设上述论断对 n 已成立，讨论 $n+1$ 的情况．按归纳假设，KS 以 t_n 为前缀．利用在 §10.4 中的结果(或参见附录 C)，KS 在 t_n 之后不是 t_n 就是 $\hat{t}_n$．如同时存在周期为 2^{n+1} 的轨，将其踪迹记为 x^∞，$|x|=2^{n+1}$，且不妨设 $x=M(x)$；那么容易证明 $x\geqslant t_n\hat{t}_n=t_{n+1}$．这就可以推出揉序列一定是以 t_{n+1} 为前缀的．归纳证明已经完成．

现在容易解决我们的问题．如果存在周期为 2^n 的轨，但不存在周期为 2^{n+1} 的轨，则由上可见 $\mathrm{KS}=t_n^\infty$．如果存在周期为所有 $2^n(n\geqslant0)$ 的周期轨，但不存在任何其他周期的周期轨，那么根据上

述证明，揉序列 KS 将以每一个 t_n 为它的前缀. 因此就推出

$$\mathrm{KS}=t_{\infty}$$

(参见 § 10.2 关于 t_{∞} 的定义). 在 § 13.3 中已经证明, 这两种情况的熵都是 0.

§ 14.6 熵与揉序列

从熵的定义开始, 我们就知道语言 $\mathscr{L}(\mathrm{KS})$ 的熵是由揉序列确定的. 从本节的讨论已经看到,熵虽然只是一个数值,但它往往反映出动力系统的重要动力学行为. 从 § 14.4 对于周期窗口的熵的讨论, 说明在这时从熵的数值只能确定出动力系统的动力学行为处于那一个周期窗中,而不能反映出进一步的细节.

在这方面再引用[21]中的一个结果.

定理 如果单峰映射的增长数 s 不是形式为 $s^n \pm s^{n-1} \pm \cdots \pm 1=0$ 这样的方程的根, 那么, 揉行列式 $D(t)$ 是由 s 唯一确定的.

对于这个定理的应用还不很清楚,但它说明,在一定条件下熵 h 唯一地决定了揉序列以及形式语言本身.

§ 15 禁止字与正规语言

如果说熵反映的是在语言中允许出现的符号串的多样性, 那么禁止字则是从在语言中不允许出现的符号串来看问题. 对于具有 § 7.4 中两个性质的语言, 在本节中定义禁止字及其集合 L″. 对于单峰映射中的两类正规语言, 研究了它们的禁止字集合的不同特点.

§ 15.1 关于禁止字的一般概念

设 S 为有限符号集, $\mathrm{L}\subseteq S^*$ 是在 S 上的一个形式语言. 我们定义另一个形式语言

$$L' = S^* - L,$$

称之为语言 L(在 S^* 中)的补. 每一个符号串 $z \in L'$ 都不允许在 L 中出现.

容易理解, 知道了 L' 也就等于知道了 L, 因此 L' 也是对 L 的一个刻划. 但是一般来说, 通过 L' 来研究 L 并非是有效的途径. 在乔姆斯基的层次理论中, 正规语言类 $\mathscr{L}$(RG) 关于取补是封闭的, 但上下文无关语言类 $\mathscr{L}$(CF) 和递归可枚举语言类 $\mathscr{L}$(RE) 则关于取补运算不封闭. 上下文有关语言类 $\mathscr{L}$(CS) 关于取补运算封闭是最近才得到证明的结果.

现在假定我们研究的语言 L 总是具有在 §7.4 中提出的两个基本性质:

性质 1 $z \in L \Leftrightarrow z$ 的每一个子串 $z' \in L$.

性质 2 $z \in L \Rightarrow$ 至少存在一个 $a \in S$, 使符号串 za 也属于 L.

在 § 7.4 中已经说明, 单峰映射的形式语言 $\mathscr{L}$(KS) 一定具有这两个性质, 并且解释了它们的动力学意义. 这两个性质在第 8、9 节的讨论中起了极其重要的作用. 在下一章中, 我们将会看到在元胞自动机研究中出现的形式语言也具有这两个性质. 基于它们的动力学含义, 我们给出

定义 称具有上述两个性质的形式语言为 D 类语言. (这里的字母 D 是动力系统的英文的第一个字母.)

对于 D 类语言 L, 研究补 $L' = S^* - L$ 往往会有方便之处. 实际上, 如 $z \in L'$, 则 z 不属于 L, 同时, 以 z' 为子串的任何符号串也不属于 L, 从而就在 L' 中. 这是反面利用性质 1 的结果.

由此出发, 可以对 D 类中的语言 L 提出"禁止字"概念, 它是对语言的一种新的刻划.

如果 $z \notin L$, 但 z 的任何真子串都属于 L(在 [53] 中称为 Distinct Excluded Block), 则称符号串 $z \in S^*$ 为语言 $L(\subseteq S^*)$ 的禁止字.

如在§8.1中指出的那样，性质1已隐含了空串$\varepsilon \in$L成立。因此，如L为D类语言，则$\varepsilon \notin$L'。当然，ε不是L的禁止字。

由上面的定义可见，语言L的禁止字即是不属于L而长度达到极小的符号串。这里极小的含义是，从禁止字的开始或结尾去掉一个或多个符号之后，留下的子串已属于L。

我们把禁止字全体叫做语言L'的核，记为L''。

容易看出，在L、L'和L''之间的基本关系为

$$L' = S^*L''S^*,$$

$$L = S^* - L' = S^* - S^*L''S^*.$$

L''的一个基本性质是，如果x、$y \in$ L''，$x \neq y$，那么其中的任何一个符号串不能以另一个符号串为自己的子串。

以上只利用了D类语言所具有的第一个性质。从性质2可以知道，如果$S = \{0, 1\}$，$z \in$ L''，那么一定有$\hat{z} \in$ L。也就是说，将z的最后一个符号取反码，其他保持不动，所得的符号串$\hat{z}$一定是语言L的字。

现在举几个简单例子，其中均取$S = \{0, 1\}$，L为单峰映射的形式语言。

[例1]　$KS = 10^\infty$。

这时$L = S^*$，$L' = L'' = \varnothing$。

[例2]　$KS = 0^\infty$。

在§7.2已知$\mathscr{L}(0^\infty) = \{0^n | n \geqslant 0\}$。L'为无限语言，但它的核则极其简单：$L'' = \{1\}$。

[例3]　$KS = 1^\infty$。

在§7.2已知$\mathscr{L}(1^\infty) = 0^*1^*$。禁止字集合L''则仍是很简单的：$L'' = \{10\}$。实际上，它表明在写符号串时，只要在符号1后不接着写0，就得到L中的字。

[例4]　$KS = (10)^\infty$。

接受$\mathscr{L}((10)^\infty)$的最小自动机见图9-2。在§13.3中计算了它的熵，同时还给出了这个语言的正规表达式。用L''来刻划L

是非常简单的:

$$L''=\{100,\ 1011\}.$$

[例 5]　$KS=(101)^\infty$.

这个例子在 § 9.1 已作过详细讨论. 在语言 $\mathscr{L}((101)^\infty)$ 中反映了任意周期的周期轨的存在性,同时还存在所谓李-约克意义上的混沌集合. 实际上,接受 L 的最小有限自动机的主要部分与[18]中的有向图是相同的. 用 L″ 来刻划 L 在这里也非常简单:

$$L''=\{100\}.$$

[例 6]　$KS=(100)^\infty$.

图 9-2 中作出了接受语言 $L=\mathscr{L}((100)^\infty)$ 的最小有限自动机. 这时,有

$$L''=\{1000,\ 10011,\ 100101\}.$$

[例 7]　$KS=101^\infty$.

这个例子在 § 9.4 和 § 14.3 中都作过讨论. 与以上例子不同,这时的禁止字集合是无限集,但具有简单的结构:

$$L''=\{10(11)^n0\}_{n\geq 0}.$$

关于从语言 $L=\mathscr{L}(KS)$ 计算 L″ 的一般方法将在下面讨论. 在这一节中研究 L 为正规语言时, L″ 所具有的一般规律. 下一节将讨论 L 为非正规语言时的情况. 在这里,我们指出,因为 L″ 完全唯一地确定了语言 L,而且其结构往往比较简单,所以,有可能成为刻划 L 的一个很有希望的手段.

§ 15.2　有限补语言

如果 L 为 D 类语言,同时它对应的 L″ 为有限语言,那么,可以证明 L 为正规语言.

对于 L″ 为有限语言的情况,我们称 L 为**有限补语言**(Finite Complement Language). 因此,这一小节即是要证明,有限补语言一定是正规语言. 在上一小节的例子中,除了最后一个之外,都

是有限补语言.

采用[53]中的方法，我们描述上述命题的证明过程，并举一个例子.

设 $L \subseteq S^*$，S 为由 k 个符号组成的有限集. 如 $L=S^*$，则不必再讨论. 现设 $L \subsetneq S^*$ 为有限补语言. 由于 L'' 只含有限个符号串，可设其中最长的禁止字的长度为 b. 取长度为 $b-1$ 的所有可能符号串，一共是 k^{b-1} 个. 先将其中不属于 L 的那些去掉，其中包括禁止字，以及含有禁止字为子串的那些符号串.

将余下的每个符号串作为结点，我们可以构造一个有向图，其方法如下. 设某个结点对应于符号串 $a_1a_2\cdots a_{b-1}$. 考虑从这个结点出发的 k 条弧，每一条弧标以 S 中的不同符号. 如果标号 a 使 $a_1a_2\cdots a_{b-1}a \notin L$，则删去这一条弧. 否则，设 $a_1a_2\cdots a_{b-1}a \in L$，我们就将这条弧通向对应于 $a_2\cdots a_{b-1}a$ 的那个结点. 对每个结点都这样做，这样就构成一个有向图.

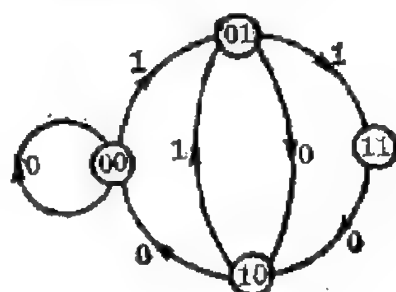


图 15-1

图 15-1 是由 $L''=\{111\}$ 所得到的上述有向图. 这里 $b=3$，$k=2$，有四个结点. 根据上述方法，从结点 11 出发只有一条标以 0 的弧，从其他结点出发则都有两条弧.

从图 15-1 容易看出，从任一结点出发，取图中的任何一条路径，都可以生成 L 中的一个字. 反之这也成立. 虽然这个有向图与 §1.4 中的有限自动机的状态转移图似乎并不完全一样，但再增加一些结点和弧，其中包括一个初始结点和带 ε 转移的弧，就可以从图 15-1 得到接受 L 的一个有限自动机，从而也就证明了 L 为正规语言.

对于 S 含 k 个符号和 L'' 中字的最大长度为 b 的一般情况，讨论的过程完全一样. 从而我们知道，有限补语言必是正规语言. 更一般的讨论见 §16.1.

§ 15.3 禁止字的计算方法

现在限于讨论 $L=\mathscr{L}(KS)$. 我们将建立从揉序列 KS 出发直接求出禁止字的计算方法.

与 § 15.1 中一样,采用记号 $\hat{z}$ 是很方便的.

现设 $\hat{z}$ 是语言 $L=\mathscr{L}(KS)$的一个禁止字. 从 § 15.1 中已知, $z\in L$ 成立. 因此成立不等式

$$z\leqslant KS.$$

另一方面, $\hat{z}$ 的每一个真后缀都属于 L, 因此可推出

$$\hat{z}>KS.$$

否则, $\hat{z}$ 的每一个后缀都小于等于 KS, 从 § 7.5 中的判定法则可推出 $\hat{z}\in L$, 引出矛盾.

合并以上两个不等式, 可以看出 z 一定是揉序列 KS 的一个前缀, 而且是偶串.

于是, 我们已经证明, 语言 $L=\mathscr{L}(KS)$的每一个禁止字是奇串, 而且在将它的最后一个符号取反码之后, 得到的符号串是揉序列的一个偶前缀.

当然, 这只是一个符号串为 L 的禁止字的必要条件, 并不是充分条件. 例如, 如

$$KS=(101)^{\infty},$$

那么 $z=101101$ 是它的一个偶前缀. 这时, $\hat{z}=101100\notin L$, 但不是 L 的禁止字. 实际上, $\hat{z}$ 的后缀 100 是 L 的(唯一)禁止字.

应用在 § 7.6 中的前后缀概念, 容易证明, 如果 z 是 KS 的偶前缀, 而且 z 不存在关于 KS 的任何偶真前后缀, 则 $\hat{z}$ 是 $L=\mathscr{L}(KS)$的禁止字. 实际上, 这是 $\hat{z}\in L''$ 的充分必要条件. 证明如下.

设 z 为 KS 的偶前缀, 则已有 $\hat{z}>KS$ 成立, 因此 $\hat{z}\notin L$. 如果 $\hat{z}$ 不是禁止字, 则由于 $\hat{z}$ 与 z 只在最后一个符号上有不同, 而 $z\in L$, 因此 $\hat{z}$ 含有一个真后缀为禁止字, 我们记它为 $\hat{z}'$. 这时 z' 是

KS的偶前缀，又是 z 的真后缀，因此已经找到了 z 关于KS的一个偶真前后缀。另一方面，如果 z 有一个偶真前后缀 z'，那么 $\hat{z}'$ 为奇串，因此 $\hat{z}'>\mathrm{KS}$。这表明 $\hat{z}'\notin\mathrm{L}$。因此以 $\hat{z}'$ 为真后缀的符号串 $\hat{z}$ 不会是L的禁止字。

上述条件也等价于：z 为KS的偶前缀，同时 z 是循环移位最大字。证明从略。

利用以上结果，从揉序列KS出发很容易直接确定出语言 $\mathscr{L}(\mathrm{KS})$ 的全部禁止字。

§15.4 KS为周期序列时的禁止字

按照[53]，当L″为有限语言时，我们称L为有限补语言。与此相对应，当L″为无限语言时，我们称L为**无限补语言**(Infinite Complement Language)。根据§15.2中的结果，任何非正规语言必是无限补语言。但是它的逆命题并不成立。

现在讨论 $\mathscr{L}(\mathrm{KS})$ 为单峰映射的情况。

从§8的定理已知，这时只有两种可能，即KS为周期序列或终极周期序列。

如果 $\mathrm{KS}=x^\infty$，我们容易证明，语言 $\mathrm{L}=\mathscr{L}(\mathrm{KS})$ 必是有限补语言。这个证明所需要的准备知识写在附录B.1中。在那里证明了：如 x 为偶串，则 $z\in\mathscr{L}(x^\infty)$ 的充分必要条件是 z 的每一个长度不超过 $|x|$ 的子串 $z'\leqslant x$。这直接推出语言 $\mathrm{L}=\mathscr{L}(x^\infty)$ 的禁止字的长度，当 x 为偶串时不会超过 $|x|$，而在 x 为奇串时不会超过 $2|x|$。

在§15.1中的例2到例6即是上述结论的具体例子。

于是，$\mathrm{L}=\mathscr{L}(x^\infty)$ 的禁止字只可能为有限多个。

§15.5 KS为终极周期序列时的禁止字

在§15.1中的例1为 $\mathrm{KS}=10^\infty$，这时没有任何禁止字。而例7，即 $\mathrm{KS}=101^\infty$，却告诉我们存在无限多个禁止字。这里的规律

是什么呢?

在[27]中已经证明，在所有终极周期而不是周期的揉序列中，$\mathscr{L}(10^\infty)$是仅有的例外．这就是说，如果 KS 为终极周期序列，但不是周期序列，也不是 10^∞，那么 $\mathscr{L}$(KS)一定是无限补语言．

这里只介绍从 $\mathrm{KS}=\rho\lambda^\infty$ 寻找无限多个禁止字的具体方法，以及 L″ 的结构特征，具体的证明细节从略。

将揉序列按以下要求写成

$$\mathrm{KS}=\rho\lambda^\infty,$$

其中 λ 为偶极小串(即长度最小的偶串)，同时满足 $\lambda=M(\lambda)$，又要求 ρ 不以 λ 为后缀．由于 KS 不是周期序列，这是可以实现的．

分两种情况讨论．

如果 ρ 为偶串，则当 k 足够大时，可以证明

$$\rho\lambda^k\hat{\lambda}\in \mathrm{L}''.$$

如果 ρ 为奇串，先求出符号串 λ 关于 KS 的最长前后缀(参见 § 7.6)，记为 λ'．可以证明，λ' 是奇串，而且当 k 足够大时，

$$\rho\lambda^k\hat{\lambda}'\in \mathrm{L}''.$$

这里条件 $\mathrm{KS}\neq 10^\infty$ 起了作用，它使 λ 不会是 0 串．又由于 KS 不是周期序列，因此 λ 不会是 KS 的前缀．这时可以应用 § 7.5 中的结果．

需要指出，上述方法求出的未必是 L 的全部禁止字．关于 L″ 的结构需要另外讨论．现在举一些例子．

[例 8]　$\mathrm{KS}=101(10)^\infty$.

令 $\rho=101$, $\lambda=1010$，即为上述第一种情况．实际上，可求出

$$\mathrm{L}''=\{100\}\cup\{101(10)^{2n}1011\}_{n>0}.$$

[例 9]　$\mathrm{KS}=100(011)^\infty$.

令 $\rho=10001$, $\lambda=101$，它也属于第一种情况．实际上，这时的

$$\mathrm{L}''=\{10001(101)^n100\}_{n>0}.$$

[例 10]　$\mathrm{KS}=100(101)^\infty$.

这时 $\rho=100$ 是奇串，属于第二种情况．求出 $\lambda=101$ 关于

KS 的最长前后缀为 $\lambda'=1$. 这对应于一族禁止字

$$\{100(101)^n 0\}_{n\geq 0}.$$

但实际上还存在第二族禁止字

$$\{100(101)^n 11\}_{n\geq 0}.$$

可以证明，当 $\mathscr{L}(\rho\lambda^\infty)$ 为无限补语言时，禁止字可分成两类. 第一类由有限族组成，每一族具有形式 $\{\rho\lambda^n\hat{\lambda}_1\}_{n\geq n_0}$，其中 λ_1 为 λ 的某个前缀. 第二类则由有限个散见的禁止字组成.

再举一个较复杂一些的例子.

[例 11]　$KS=1000101(1001)^\infty$.

利用 § 15.3 的计算方法，容易求出

$$\begin{aligned} L'' = &\{10000,\ 100011\} \cup \{1000101(1001)^n 0\}_{n\geq 0} \\ &\cup \{1000101(1001)^n 11\}_{n\geq 0} \\ &\cup \{1000101(1001)^n 101\}_{n\geq 0}. \end{aligned}$$

借用在 § 3.5 中的名词，我们也称以上 L″ 具有半线性结构.

这样，我们已经证明，§ 15.2 中论断的逆命题不成立. 也就是说，虽然有限补语言必为正规语言，但正规语言可以是无限补语言. 对于 $\mathscr{L}$(KS) 为正规语言的两种情况，从 L″ 来刻划它们的方法揭示了它们之间的不同点. 也就是说，在示意图 9-6 中的两端虽都属于复杂性层次中的最低一层，但实际上仍存在本质差异.

§ 16　禁止字与非正规语言

本节首先讨论 L 和 L″ 在乔姆斯基层次体系中所处层次之间的关系. 然后，对于在单峰映射的几个已知的非正规语言，分别计算了它们的禁止字结构. 材料主要来自作者正在进行的工作.

§ 16.1　L 和 L″ 的乔姆斯基层次

设 L 为 D 类语言，L″ 为 L 的禁止字集合. 如过去一样，记 $\mathscr{L}$(RG) 为正规语言类. 那么，容易证明：

$$\mathrm{L}\in\mathscr{L}(\mathrm{RG}) \Leftrightarrow \mathrm{L}''\in\mathscr{L}(\mathrm{RG}).$$

实际上，从关系式 $\mathrm{L}=S^*-S^*\mathrm{L}''S^*$ 出发，利用语言类 $\mathscr{L}(\mathrm{RG})$ 关于取补运算和连接运算封闭(参见§1.9)，就可以从 $\mathrm{L}''\in\mathscr{L}(\mathrm{RG})$ 推出 $\mathrm{L}\in\mathscr{L}(\mathrm{RG})$。

另一方面，再利用§1.9中语言类 $\mathscr{L}(\mathrm{RG})$ 关于镜象运算和 MIN 运算封闭，就可以推出另一半结论．说得详细一点，从 $\mathrm{L}\in\mathscr{L}(\mathrm{RG})$，先有 $\mathrm{L}'=S^*-\mathrm{L}\in\mathscr{L}(\mathrm{RG})$。然后，定义

$$\mathrm{L}_1=\mathrm{MIN}(\mathrm{L}')\in\mathscr{L}(\mathrm{RG}),$$

容易看出，$\mathrm{L}''\subseteq\mathrm{L}_1$．然后，又有

$$\mathrm{L}_2=\mathrm{L}_1^R\in\mathscr{L}(\mathrm{RG}),$$

这里 L_2 是 L_1 的镜象．可以证明，有

$$\mathrm{L}''=(\mathrm{MIN}(\mathrm{L}_2))^R,$$

因此就推出 $\mathrm{L}''\in\mathscr{L}(\mathrm{RG})$．

这同时也证明了，L 为非正规语言的充分必要条件是 L″ 为非正规语言。

可以设想，上述关于 L 和 L″ 同属 $\mathscr{L}(\mathrm{RG})$ 的结论也许可以推广到乔姆斯基层次体系中的其他每一个层次．由于 $\mathscr{L}(\mathrm{CF})$ 关于补运算与 MIN 运算不具有 $\mathscr{L}(\mathrm{RG})$ 的性质，不是不封闭就是还不知道是否封闭，因此上述证明方法失效．已经证明，L 和 L″ 同属 $\mathscr{L}(\mathrm{CS})$ 和递归函数类的结论成立．但对于 $\mathscr{L}(\mathrm{RE})$，如 L 和 L″ 都不是递归语言，则结论一定不成立．

但是对于单峰映射生成的语言 $\mathrm{L}=\mathscr{L}(\mathrm{KS})$，情况要清楚得多。

我们先证明，如果这时的 L″ 为上下文无关语言，那么它一定也是正规语言。

这里的主要工具是在§3.5中介绍的第一个定理：如果 L 是一个上下文无关语言，即存在一个上下文无关语法 G，使 $\mathrm{L}=\mathrm{L}(G)$．如果 G 具有非自嵌套性质，则 L 为正规语言．

这里的非自嵌套性质是在派生过程中不出现 $A\Rightarrow\alpha A\beta$，其

中 A 为变量，α 和 β 是由变量和终结符组成的非空串.

现设 $\mathrm{L}=\mathscr{L}(\mathrm{KS})$，$\mathrm{L}''$ 为上下文无关语言. 这时存在一个上下文无关语法 G，使

$$G=(V, S, P, s_0), \qquad \mathrm{L}''=\mathrm{L}(G).$$

按照[1]，可以假设 G 的每一个变量 A 都是所谓有用变量，即存在派生

$$s_0 \overset{*}{\Longrightarrow} wAx \overset{*}{\Longrightarrow} \gamma,$$

其中 w、$x\in(V\cup S)^*$，$\gamma\in S^*$. 又不妨假设 w 和 x 中已不含变量.

如果上述 G 已具有非自嵌套性质，则不必再讨论. 否则，设存在一个 $A\in V$，使

$$A \overset{*}{\Longrightarrow} \alpha A\beta,$$

且非空串 α 和 β 中也已不含变量. 由于 A 为有用变量，存在派生

$$s_0 \overset{*}{\Longrightarrow} wAx \overset{*}{\Longrightarrow} w\alpha A\beta x,$$

于是就有 $s_0 \overset{*}{\Longrightarrow} w\alpha^n A\beta^n x$ 对一切 $n\geqslant 0$ 成立.

设 $A \overset{*}{\Longrightarrow} \delta$，$\delta\in S^*$，那么我们就得到在 L'' 中的一族字 $\{w\alpha^n\delta\beta^n x\}_{n>0}$.

由于在§15.3中的讨论，我们知道每个禁止字的每个真前缀都是揉序列 KS 的前缀. 由以上讨论，可见揉序列 KS 一定以 $\{w\alpha^n\}_{n>0}$ 中任何一个符号串为前缀. 从而只能是

$$\mathrm{KS}=w\alpha^\infty,$$

即终极周期序列或周期序列. 应用§8的结果，$\mathrm{L}=\mathscr{L}(\mathrm{KS})$ 为正规语言，然后，再应用这一小节一开始的结果，就证明 L'' 也是正规语言.

回顾§12.4，我们看到，虽然对于单峰映射中的形式语言 $\mathrm{L}=\mathscr{L}(\mathrm{KS})$，猜测

$$\mathrm{L}\in\mathscr{L}(\mathrm{CF}) \Rightarrow \mathrm{L}\in\mathscr{L}(\mathrm{RG})$$

尚未证明，但对于 L″ 而言，则已建立了这样的事实.

由以上讨论可知，对于非正规语言 $L=\mathscr{L}(KS)$，它的禁止字集合 L″ 在乔姆斯基层次体系中至少是上下文有关语言.

利用禁止字与揉序列之间的直接关系(见 § 15.3)，以及接受上下文有关语言的线性有界自动机(见 § 2.6)，不难证明

$$L\in\mathscr{L}(CS) \Leftrightarrow L''\in\mathscr{L}(CS),$$

这里 $\mathscr{L}(CS)$ 是上下文有关语言类.

对于更高层次的语言类，容易建立以下事实. 对一般的 D 类语言 L，L 为递归语言和 L″ 为递归语言是等价的. 对于 $L=\mathscr{L}(KS)$，L″ 为递归可枚举语言时，L 也一定是递归语言. 这里的递归语言类是在 § 2.5 中介绍过的，介于递归可枚举语言类 $\mathscr{L}(RE)$ 和上下文有关语言类 $\mathscr{L}(CS)$ 之间的一个语言类.

这里不清楚的是，对于 $L=\mathscr{L}(KS)$ 来说，是否成立

$$L\in\mathscr{L}(RE) \Rightarrow L \text{ 为递归语言?}$$

对于 L 和 L″ 中有一个属于 $\mathscr{L}(RE)$，但不是递归语言的情况，容易知道，另一个必是非递归可枚举语言. 实际上，否则将导致 L 和 L″ 同属于 $\mathscr{L}(RE)$，从而它们都是递归语言(这里所用到的事实可参看[1]～[6]). 当然，L 和 L″ 都不属于 $\mathscr{L}(RE)$ 也是可能的.

由以上分析可见，如 $L=\mathscr{L}(KS)$ 为非正规语言，则对于具体的例子来说，禁止字集合 L″ 均属于上下文有关语言. 由于目前还缺乏系统的理论，以下几节分别研究几个在第 4 章已出现过的例子.

§ 16.2 费根鲍姆吸引子的禁止字

与 § 10 一样，用记号 t_∞ 表示费根鲍姆吸引子的揉序列：

$$t_\infty=1011\ \ 1010\ \ 1011\ \ 1011\ \ \cdots\cdots.$$

利用 § 15.3 的计算方法，容易看出按长度排列的前几个禁止字是

$$100,\ 10110,\ 101111.$$

采用 § 10 的记号，记

$$h=\{0\to11,\quad 1\to10\}$$

为倍周期分岔引出的同态. 我们知道, 有

$$t_\infty=h^\infty(1),\qquad h(t_\infty)=t_\infty.$$

现在来证明语言 $L=\mathscr{L}(t_\infty)$ 的禁止字集合为

$$L''=\{h^n(100)\}_{n\geq0}\cup\{h^n(10110)\}_{n\geq0}.$$

这里的关键在于证明, 除了 100 和 10110 之外, 每一个禁止字的长度都是偶数.

现在我们假设 x 是一个奇数位长度的禁止字. 从 § 15.3 可知, $\hat{x}$ 是 t_∞ 的偶前缀, 长度也是奇数. 由于 t_∞ 的奇数位符号都是 1(见 § 10.2), 因此, x 的最后一个符号一定是 0. 这样, 就可以记禁止字为

$$x=t_n\alpha0,$$

其中 $|\alpha0|<|t_n|=2^n$, $t_n\alpha$ 是 t_∞ 的前缀.

如果子串 $\alpha\neq\varepsilon$, 则存在 $i\geq0$, 使 $\alpha=t_i\alpha'$, 其中 $|\alpha'|<|t_i|=2^i$. 因为禁止字 x 为奇串, t_n 也是奇串, 因此 α' 也是奇串. 这导致

$$\alpha'0>\alpha'1.$$

由于 α' 的长度为偶数, $\alpha'1$ 是 t_∞ 的前缀, 从而导致 $\alpha'0\notin L$. 这与 x 为禁止字相矛盾.

如果 $\alpha=\varepsilon$, 则 $x=t_n0$. 如 n 为奇数, 则容易归纳证明出每个 t_n 以 0 结尾, 从而 x 以 100 为后缀. 由于 x 为禁止字, 所以只能是 $x=100$. 如 n 为偶数, 那么同样可证明每个 $t_n(n\geq2)$ 以 1011 为后缀. 由于 x 以禁止字 10110 为后缀, 所以 $x=10110$.

于是, 如果 $x\in L''$, 但不等于 100 或 10110, 那么 $|x|$ 为偶数. 由于 $\hat{x}$ 为 t_∞ 的前缀, 奇数位均为符号 1, 因此 $h^{-1}(x)$ 有定义. 利用 h 保序, 以及 § 15.3 中关于禁止字的充分必要条件, 可见 $h^{-1}(x)$ 也是语言 $L=\mathscr{L}(t_\infty)$ 的禁止字. 如果 $h^{-1}(x)$ 仍为偶数长度, 则可以证明 $h^{-2}(x)$ 也是禁止字. 依次类推, 最后得到 $h^{-n}(x)$ 不是 100, 就是 10110, 从而 $x=h^n(100)$ 或 $h^n(10110)$. 证毕.

这时 L'' 中字长的规律很简单, 即为 $3\cdot2^n$ 和 $5\cdot2^n$, $n\geq0$.

在上一小节中已经证明，对于单峰映射的非正规语言 $L(=\mathscr{L}(KS))$，禁止字集合 L'' 不会是上下文无关语言.

对于 $L=\mathscr{L}(t_\infty)$，从 L'' 中字长的上述结构，容易直接看出 L'' 不是上下文无关语言. 实际上，如果 L'' 为上下文无关语言，则可以用同态

$$g=\{0\to 0,\ 1\to 0\}.$$

这时由语言类 $\mathscr{L}(CF)$ 关于同态封闭，知道 $g(L'')$ 也是上下文无关语言. 但 $g(L'')$ 是由一个符号生成的形式语言，应用 § 3.5 中的定理 2，$g(L'')$ 也是正规语言，同时其中的字长所成的整数集为半线性集. 由于 L'' 目前的字长集合明显不是半线性集，因此就推出 L'' 不是上下文无关语言.

容易直接证明 L'' 为上下文有关语言. 除了直接从同态 h 和线性有界自动机的定义来作出证明之外，也可以从 § 4.1 出发，用 DOL 系统生成 $\{h^n(100)\}_{n\geq 0}$ 与 $\{h^n(10110)\}_{n\geq 0}$. 将 DOL 语言看成 ETOL 语言的特例，即可推出 L'' 为 ETOL 语言，从而也是上下文有关语言.

可以将以上讨论与第 4 章中前两节对比，讨论 L'' 比讨论 L 显然容易得多.

关于 L'' 引进一个指标 α 有时是很有用的. 定义 n_k 是 L'' 中长度不超过 k 的禁止字个数，然后计算

$$\alpha=\lim_{k\to\infty}\frac{n_k}{k},$$

称 α 为 L'' 的指标. 如果极限不存在则可以取上极限. 我们可以说 α 是对 L'' 的一个刻划，当然也是对 L 的一个刻划.

应用 § 3.5 的定理 2，如果 L'' 是上下文无关语言或正规语言，且为无限语言，那么由于 L'' 的字长所成集合为半线性集，可见 α 为正常数. 换句话说，如果 L 为无限补语言，指标 $\alpha=0$，则 L'' 一定不是上下文无关语言. 当然，这只是一个充分条件.

对于 $L=\mathscr{L}(t_\infty)$，在 $k=5\cdot 2^n$ 时，$n_k=2(n+1)$. 由此容易证明

指标 $\alpha=0$.

§ 16.3 偶斐波那契系统的禁止字

现在研究在 § 12.2 中介绍的偶斐波那契系统的禁止字. 为简单起见, 只讨论在 § 12.3 中的第一个例子. 记 $\alpha=101$, $\beta=11$, 引入同态

$$h=\{\alpha\to\alpha\alpha\beta,\quad \beta\to\alpha\beta\},$$

并且记

$$e_n=h^n(\alpha),$$

就可以得到揉序列

$$e_\infty=\lim_{n\to\infty} e_n,$$

或记为 $h^\infty(\alpha)$. 在表 12.3 的第一行列出了 e_∞ 的前 32 位.

语言 $L=\mathscr{L}(e_\infty)$ 虽是由 e_∞ 所确定, 但直接从 e_∞ 出发分析 L 的结构是件不太容易的工作. 我们可以看到, L 的禁止字集合很简单. 从 $e_\infty=1011011110\cdots$已可求出最短的两个禁止字是

$$100 \text{ 和 } 10110110.$$

可以将它们记为 $\hat{e}_0(=\hat{\alpha})$和 $\hat{e}_1(=\alpha\alpha\hat{\beta})$.

我们将要证明 $L=\mathscr{L}(e_\infty)$的禁止字集合为

$$L''=\{\hat{e}_n\}_{n\geq 0}.$$

应用数学归纳法. 设 $\hat{e}_0$、$\hat{e}_1$、$\cdots$、$\hat{e}_n$ 已是 L 的禁止字, 而且 L 没有比它们中任何一个更短的禁止字. 考虑 e_∞ 的前缀 e_{n+1}, 观察按照 § 15.3 的充分必要条件在其中能生成几个新的禁止字.

写出 $\quad e_{n+1}=h^{n+1}(\alpha)=h^n(\alpha)h^n(\alpha)h^n(\beta)=e_ne_nh^n(\beta)$.

观察它的长度超过$|e_n|$的偶前缀. 因为 e_n 本身是偶串, 容易看出, 从 e_ne_n 的偶前缀不会生成禁止字. 现在将第三个因子写为

$$h^n(\beta)=h^{n-1}(\alpha)h^{n-1}(\beta)=e_{n-1}e_{n-2}\cdots e_1e_0\beta,$$

由于每个 e_i(其中 $i=0,1,\cdots,n-1$)为偶串, 而 $\beta=11$, 因此从 e_{n+1} 中只有 $\hat{e}_{n+1}$ 才可能是新的禁止字.

这里 $\hat{e}_{n+1}\notin L$ 是明显的. 为了证明 $\hat{e}_{n+1}$ 确实是禁止字, 只要

证明 e_{n+1} 不以 e_0、e_1、…、e_n 中的任何一个为后缀．写出

$$e_{n+1}=e_ne_nh^n(\beta),$$
$$e_n=e_{n-1}e_{n-1}h^{n-1}(\beta)=e_{n-1}h^n(\beta),$$

利用数学归纳法，即可完成这个证明．

现在来计算在 § 16.2 中提出的指标 α．

记符号串 $h^n(\alpha)$ 与 $h^n(\beta)$ 的长度为 a_n 和 b_n，则可以建立递推关系

$$a_{n+1}=2a_n+b_n,\qquad b_{n+1}=a_n+b_n.$$

求出系数阵的特征值，即可得到

$$\lim_{n\to\infty}\frac{a_{n+1}}{a_n}=\frac{3+\sqrt{5}}{2}\approx 2.236\cdots.$$

由于长度不超过 a_n 的禁止字个数为 $n+1$，可以看出指标 $\alpha=0$．由 § 16.2 的讨论知道，L″ 的复杂性层次至少为上下文有关语言．利用 L 系统可以证明 L″ 是 ETOL 语言，从而确是上下文有关语言．

§ 16.4 奇斐波那契系统的禁止字

只讨论在 § 12.3 中的第二个例子．

定义同态

$$h=\{0\to 110,\quad 1\to 10110\},$$

又令 $m_0=1$，$m_1=10$，$m_2=101$．然后定义 $\{m_n\}_{n\geq 0}$ 为按照

$$h(m_n)=m_{n+3}$$

生成的符号串序列．这样可以得到揉序列

$$m_\infty=\lim_{n\to\infty}m_n.$$

在这个例子中 $L=\mathscr{L}(m_\infty)$ 的禁止字集合 L″ 的计算要比前两小节的情况复杂．下面只证明它的指标 α 为 0，然后结合其他讨论可推出 L″ 为上下文有关语言．

从 $\{m_n\}$ 的定义可以建立两个基本公式：

$$m_{3l+1}=m_{3l}m_{3l-3}m_{3l-2},$$

$$m_{3l}=m_{3l-2}m_{3l-3}m_{3l-2},$$

$l\geqslant 1$. 同时，每个 m_{3l} 和 m_{3l+1} 是奇串.

先证明如 $\hat{x}$ 为 L 的禁止字，且 $|\hat{x}|\leqslant|m_{3l+1}|$，那么 $|\hat{x}|<|m_{3l}|$. 这意味着，从 m_{3l+1} 中能生成禁止字的前缀必在上述第一个公式右方的第一个因子内. 用反证法. 如果 $\hat{x}=m_{3l}\hat{u}$，$|u|\leqslant|m_{3l-3}|$，那么可写出

$$x=m_{3l-2}m_{3l-3}m_{3l-2}u.$$

这时 $m_{3l-2}u$ 是 x 的前缀，从而也是棵序列 m_∞ 的前缀. 由于 x 为偶串，u 为奇串，可推出 $m_{3l-2}u$ 是 m_∞ 的偶前缀. 应用 § 15.3 的条件，与 $\hat{x}$ 为禁止字相矛盾. 如果 $\hat{x}=m_{3l}m_{3l-2}\hat{u}$，$|u|\leqslant|m_{3l-2}|$，那么 u 也是 m_{3l} 的偶前缀，又与 $\hat{x}$ 为禁止字相矛盾.

同样可以证明，如 $\hat{x}$ 为禁止字，且 $|\hat{x}|<|m_{3l}|$，那么 $|\hat{x}|\leqslant|m_{3l-2}m_{3l-3}|$.

现在设 n_l 是语言 $L=\mathscr{L}(m_\infty)$ 的长度不超过 m_{3l+3} 的禁止字的个数，那么只要证明

$$\lim_{l\to\infty}\frac{n_l}{|m_{3l+3}|}=0,$$

就可以证明指标 $\alpha=0$.

下面分别计算数列 $\{n_l\}$ 与 $\{|m_{3l+3}|\}$ 的增长数.

分母的增长数计算是容易的. 从 $\{|m_n|\}_{n>0}$ 即为斐波那契数列 1、2、3、5、8、…，可见它的增长数为 $(1+\sqrt{5})/2$. 于是，$\{|m_{3l+3}|\}_{l>0}$ 的增长数是

$$\lim_{l\to\infty}\sqrt[l]{|m_{3l+3}|}=\frac{(1+\sqrt{5})^3}{8}\doteq 4.236\cdots.$$

可以证明 $\{n_l\}$ 的增长数为

$$\lim_{l\to\infty}\sqrt[l]{n_l}=1+\sqrt{2}\doteq 2.4142\cdots.$$

从而就可以证明所需要的结果.

为简单起见，估计出极限

$$\lim_{l\to\infty}\sqrt[l]{n_l}<3$$

也已经够了. 下面我们写出估计的方法.

从前面的讨论已知, 长度不超过 m_{3l+3} 的禁止字有两类. 第一类长度不超过 m_{3l}, 共有 n_{l-1} 个. 第二类禁止字的形式为 $m_{3l}\hat{u}$, 其中 u 是 m_{3l} 的奇前缀.

从 $m_3=m_{3l-2}m_{3l-3}m_{3l-2}$ 可以看出, 上述 u 或者在 m_{3l-2} 内, 或者超过 $m_{3l-2}m_{3l-3}$. 否则 $m_{3l}\hat{u}$ 不会是禁止字.

从 $m_{3l-2}=m_{3l-3}m_{3l-6}m_{3l-5}$ 可以看出, 也有同样的情况.

用归纳法可以将这个分析继续下去直到 m_1 和 m_0 为止.

现在取 p_l 是在 m_{3l} 中合乎上述要求的奇前缀的个数, q_{l-1} 是 m_{3l-2} 中合乎上述要求的奇前缀. 那么从上述讨论得到

$$n_l-n_{l-1}\leqslant p_l,$$

$$p_l=2q_{l-1},$$

$$q_{l-1}=p_{l-1}+q_{l-2}$$

从后两个等式得到 $q_{l-1}=3q_{l-2}$. 利用 $m_0=10$, 可以确定 $q_0=2$, 从而得到

$$q_l=2\cdot 3^{l-1},\qquad p_l=4\cdot 3^{l-1}.$$

现在从 $n_l-n_{l-1}\leqslant p_l$ 出发, 又从 $m_3=10110$ 可见只有一个偶前部 101, 因此 $n_0=1$. 这样就可估计出

$$n_l\leqslant p_l+p_{l-1}+\cdots+p_1+n_0=2\cdot 3^l-1.$$

因此得到

$$\lim_{l\to\infty}\sqrt[l]{n_l}<3.$$

这样就完成了指标的计算.

第6章

元胞自动机

本章将前几章中发展起来的方法用于与区间映射完全不同的另一类动力系统的研究.

元胞自动机可以看成为无穷维动力系统中的一类，其特点是空间、时间和状态都离散，同时每一个变量只取有限多个状态.

本章共有五节，即从§17到§21. 在§17中介绍了基本概念和动力学行为的四类现象. §18介绍了数学记号与几个基本事实，它们对于元胞自动机的研究均是必要的知识. §19、§20分别分析了语言的复杂性，其中利用了最小自动机、禁止字以及在§1中介绍的几乎全部概念. 在最后一章介绍拓扑熵及其应用.

本章完全是引论性质的介绍. 大量的未解决问题为这个困难而有趣的领域展现了广阔的前景.

§17 元胞自动机的基本概念

本节是关于元胞自动机的一般性介绍. 材料主要取自[92]、[93].

§17.1 一维元胞自动机

设在一条直线上按等间隔方式分布着完全相同的一系列元胞(Cell). 抽象地可以认为这里的每个元胞是具有一定功能的信息处理机或自动机，它们的功能将在下面介绍.

每一个元胞的状态只有有限多个. 对于只有两个状态的情况，我们与本书其余部分一样，用符号0和1来表示它们.

假设上述直线在两个方向上都没有限制，因此就有无限多个元胞．所有元胞的状态全体可以用双侧无限的符号序列表示出来．我们称每一个这样的序列为元胞自动机的一个构形（Configuration）．为了确定起见，可以将分布在直线上的元胞位置与整数全体对应起来．特别是将与整数 0 对应的位置称为基点．如记构形为 a，则可以表示成

$$a=(\cdots a_{-1}a_0a_1\cdots),$$

其中 a_0 是在基点上的元胞状态，其余依次类推．

每个元胞的状态看成是一个变量，它只能取有限个值，即有限个状态．于是上述元胞自动机即是有无限多个变量的系统．

假设时间也是离散化的．用 t 表示时间，它取整数值．称 $t=0$ 为初始时刻，它的下一时刻为 $t=1$，$\cdots$．

所有元胞的状态是同时发生变化的．记在时刻 t 的构形为 a^t，那么在时刻 $t+1$ 的构形 a^{t+1} 完全由 a^t 决定．同时，在时刻 $t+1$ 的第 i 个元胞的状态是由时刻 t 的第 i 个元胞以及相邻的距离不超过 r 的 $2r$ 个元胞的状态所决定的．用公式写出来即是

$$a_i^{t+1}=f(a_{i-r}^t,\ \cdots,\ a_{i-1}^t,\ a_i^t,\ a_{i+1}^t,\ \cdots,\ a_{i+r}^t),$$

其中的映射 f 与 i 和 t 都无关．我们称 r 为邻域半径，称 f 为元胞自动机的**局部映射**或**局部规则**．

记 S 为有限状态集．由 S、r 和 f 完全确定了一个元胞自动机．f 是从 S^{2r+1} 映入 S 的函数．

从一个给定的初始构形 a 出发，记 $a^0=a$，就可以得到 a^1，a^2，$\cdots$．因此可以将元胞自动机作为动力系统来研究．

以下介绍最简单的初等元胞自动机．

设 $S=\{0,\ 1\}$，$r=1$，这时局部映射为

$$a_i^{t+1}=f(a_{i-1}^t,\ a_i^t,\ a_{i+1}^t).$$

这时 f 是从 S^3 映入 S 的函数，自变量只有八种可能组合．只要给定 f 在这八个自变量组合上的值，f 就完全确定了．图 17-1(i) 即是一个例子，在第一行列出了从 111 到 000 的八种组合，

(i)	111	110	101	100	011	010	001	000
	0	1	0	0	1	1	0	0

(ii) 010111110101011100010

1010001010101010001

图 17-1

在每种组合的下方标出 f 的值. 在(ii)中的第一行代表某个构形中的一段,第二行则是按照 f 从第一行得到的符号串,它在两端各少一个符号,因为那要取决于第一行两端尚未写出的符号.

由此可见,$S=\{0, 1\}$ 和 $r=1$ 时的 f 只有 $2^8(=256)$ 种可能性. 它们对应于 256 种不同的元胞自动机, 通常称为**初等元胞自动机**.

由于在 S 中具体采用什么符号并不重要, 这里重要的是 S 所含符号的个数 k. 因此,上述初等元胞自动机也称为 $k=2$ 和 $r=1$ 的元胞自动机.

现在介绍初等元胞自动机的编号方法. 以图 17-1(i)为例,将 f 的八个值按从左到右的顺序看成为一个八位的二进制数 01001100,然后计算它的十进制值,得到

$$1\times64+1\times8+1\times4=76,$$

这就是这个元胞自动机的编号. 今后称它为 76 号初等元胞自动机. 反之,对于从 0 到 255 中的任何一个十进制数,容易将它改写成八位的二进制数 $\alpha_7\alpha_6\cdots\alpha_1\alpha_0$, 然后按照 $111\to\alpha_7$, $110\to\alpha_6$, $\cdots$, $000\to\alpha_0$ 定义局部映射 f. 这样就完全确定了一个初等元胞自动机.

§17.2 几种推广

可以研究分布在半无限直线上的元胞自动机. 这时将它的一个端点记为 $i=0$,而用所有非负整数表示元胞位置. 在局部映射 f 中, 自变量只出现 a_i、a_{i+1}、$\cdots$、a_{i+r}. 我们称这样的元胞自动机为**单向(One-Way)元胞自动机**.

如果局部映射的形式为

$$a_i^{t+1}=f(a_{i-r}^t+\cdots a_i^t+\cdots+a_{i+r}^t),$$

即在邻域中每个元胞的状态都起同样作用，则称为完全(Totalistic)元胞自动机。

可以证明，一般的元胞自动机的动力学行为一定可以用某个适当的单向元胞自动机或完全元胞自动机来模拟(见[92]、[93])。

在用元胞自动机模拟某些物理现象时，往往会对局部映射提出某些附加限制。常用的有：

1° 如将符号0看成为静止状态，则要求

$$f(0, 0, \cdots, 0)=0.$$

2° 除了 f 与 i 无关这个反映空间齐性的条件之外，还要求 f 是对称的，即

$$f(a_{i-r}, \cdots, a_i, \cdots, a_{i+r})=f(a_{i+r}, \cdots, a_i, \cdots, a_{i-r}).$$

它是空间各向同性的反映。

称满足这两个条件的元胞自动机为合法(Legal)元胞自动机。在 $k=2$ 和 $r=1$ 的256种初等元胞自动机中，有32种是合法的，其中包括图17-1(i)的76号在内。

以上介绍的都是一维元胞自动机。当然可以研究高于一维的元胞自动机。其中最著名的一种是在七十年代由康威(J.H. Conway)提出的"生命游戏"(Game of life)。下面作些简单介绍。

生命游戏是一个二维元胞自动机。设想将平面划分成方格棋盘，每一个方格代表一个元胞自动机，它有两个状态，用符号0表示死亡，用符号1表示活着。每个元胞在下一时刻的状态(死或活)是由当前时刻的这个元胞和与它邻接的八个元胞的状态所决定的。

具体规则如下。如果一个元胞处于活着的状态，则在八个相邻元胞中有两个或三个活着时，在下一时刻继续活着，否则就死亡。这个规则模拟生命在过分拥挤或孤单时不能生存下去。另一方面，如一个元胞处于死亡状态，但在邻域中有三个元胞处于活着的状态，则下一时刻这个元胞的状态变成活着，否则仍保持死亡状

态. 可以用计算机来模拟这些规则，也容易利用电子技术做成游戏机. 有趣的是，用这些简单规则定义的这个元胞自动机能够模拟生命活动中的生存、灭绝、竞争等等复杂现象. 康威还证明，这个元胞自动机具有通用图灵机的计算能力(参见§2.4和[1]).

关于这里介绍的生命游戏，可参看[15]、[94].

§17.3 元胞自动机的一般特征

这里列举以下特征.

空间离散、齐性 这反映在每个元胞的变化都服从相同的规律，元胞的分布方式相同.

时间离散 这与前几章一样.

状态离散、有限 这与前面的迭代映射不同. 在那里的状态是连续的，只有经过粗粒化处理后才能转化成符号序列. 对于元胞自动机不存在需要粗粒化处理的问题.

同步计算 可以将元胞自动机的构形变化看成为是对数据或信息的计算(或处理). 这时的重要特征是计算的并行性. 这与图灵机很不一样. 在第1章中介绍的各种自动机，包括图灵机在内，都有专门的读入头或读写头，每一个时刻完成一步动作，即完全按串行方式工作. 元胞自动机的工作方式与§4中的并行重写系统有相同之处.

局部性 每一个元胞的当前状态，只对于半径r的邻域中的元胞在下一时刻的状态可能发生影响. 从信息传输的角度来看，在元胞自动机中的信息传输速度是有限的.

维数高 在动力系统中一般将变量的个数称为维数. 例如，将区间映射生成的动力系统称为一维动力系统，将平面映射生成的动力系统称为二维动力系统. 对于由偏微分方程描述的动力系统则称为无穷维动力系统. 从这个角度来看，元胞自动机是一类无穷维动力系统. 在具体应用中或在计算机模拟时当然不可能真的处理无限多个变量，但一般也总是处理数量很大的元胞组成的系

因此可以说维数高是元胞自动机研究中的一个特点.

§17.4 动力学行为的分类

如前面所指出,可以将元胞自动机看成为动力系统,从而可以将初始点、轨道、不动点、周期轨和终极周期轨等一系列概念用到元胞自动机的研究中来. 但目前的“点”乃是一个构形,即一个双侧无限的符号序列,它本身可以有结构.在区间映射中的区间对应于这里由所有构形组成的集合,今后常称之为(构形)空间.

沃尔夫勒姆(S. Wolfram)在大量的计算机实验的基础上,提出可以将所有元胞自动机的动力学行为归纳为四类:

1. 趋于一个空间平稳的构形. 这里空间平稳即指每一个元胞处于相同状态.

2. 趋于一系列简单的稳定结构或周期结构.

3. 表现出混沌的非周期行为.

4. 出现复杂的局部结构,或者说是局部性的混沌,其中有些会不规则地传播.

这种分类不是严格的数学定义(参见[92]中有关讨论). 大致来说,前三类行为相当于低维动力系统中常见的不动点、周期轨与混沌,而第四类行为则可以与生命系统等复杂系统中的自组织现象相比拟.

在图 17-2 中给出了六个元胞自动机的计算机实验结果,其中包含了上述四类行为. 每一个图的第一行表示初始时刻 $t=0$ 的构形中的有限部分,它是随机选取的. 其余各行均是从上一行按照所标出的局部规则计算所得的结果. 在图形中用白格代表符号 0,用黑格代表符号 1. 在计算机上运行时对两端采用周期边界条件,这相当于将有限多个元胞布置在一个圆周上来应用局部规则.

在六个图中有五个是初等元胞自动机,在每个图的下边写出了它们的规则编号以及相应的八位二进制数. 由于已知在初等元胞自动机中不出现上述第四类行为,在图 17-2 的右下图选用了一

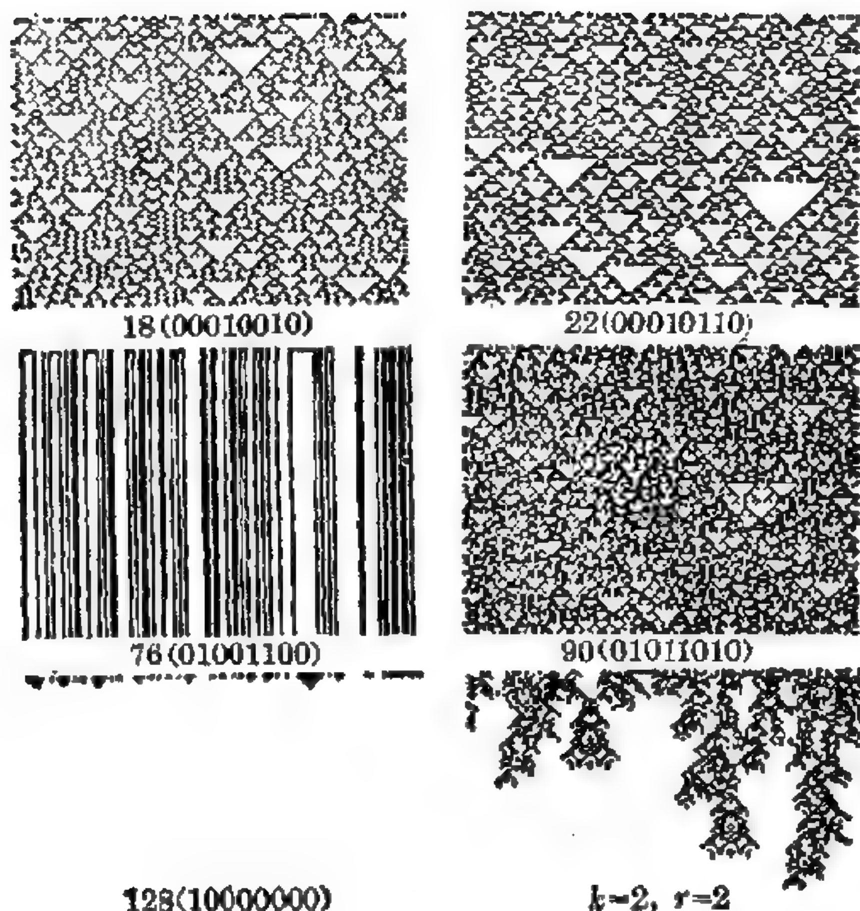


图 17-2

个 $k=2$ 和 $r=2$ 的完全元胞自动机，它的局部映射 f 在自变量为 2 和 4 时取值 1，在其他情况均取值 0.

按照沃尔夫勒姆的观点，众多(也许所有)的元胞自动机的动力学行为可归纳成数量如此之少的四类，这是非常有意义的发现．它反映出这种分类可能具有某种普适性．很可能有许多物理系统或生命系统可以按这样的分类方法来研究，尽管在细节上可以不同，但每一类中的行为在定性上是相同的．关于分类问题的重要研究可参看[93]中的论文．

§17.5 文献简述

元胞自动机是冯·诺伊曼(J. von Neumann)最早提出来的，用于模拟生命系统所具有的自复制功能(参见[95])．此后元胞自

动机还被用于模拟其他的物理系统和自然现象. 虽然从偏微分方程的数值计算方法可以生成某些元胞自动机, 但有很多元胞自动机看来并不一定是某个连续系统的离散化描述. 由于70年代以来动力系统理论的发展, 沃尔夫勒姆等人开始将动力系统方法用于元胞自动机的研究中. 如果说偏微分方程是空间连续、时间连续和状态连续的无穷维动力系统的数学模型, 那么元胞自动机则是另一类完全离散化的无穷维动力系统. 它们二者恰好成为两个极端.

在[53]中沃尔夫勒姆第一次提出将计算理论和形式语言方法用于研究元胞自动机. [92]收集了到1986年为止的重要论文,并附有大量计算机实验的结果. [93]是1989年关于元胞自动机的一次学术会议的论文集, 其中[96]、[97]对于计算理论和形式语言在元胞自动机中的研究情况作了综述.

关于元胞自动机的计算能力已有不少研究. 已知每一个图灵机都可以用元胞自动机来模拟. 已经构造出有十四个状态(即 $k=14$)的通用元胞自动机,它可以模拟任何一个元胞自动机在任何初始构形时开始的动力学行为.

§18 一些数学记号与结果

为了用形式语言来研究元胞自动机, 我们在本节介绍必要的一些数学记号, 同时列举有关的一些基本结果. 材料主要取自[96]～[98].

§18.1 构形空间与极限集

记 S 为 k 个符号的有限集, Z 为整数全体的集. 利用集合论记号,记 S^Z 是从 Z 映射到 S 的映射全体,也就是用 S 中的符号组成的双侧无限的符号序列的全体. 换句话说, S^Z 就是一维元胞自动机的构形全体所成的集合,我们称之为构形空间. 对于 $a\in S^Z$,

与上节一样记为

$$a=(\cdots a_{-1}a_0a_1\cdots),$$

其中 a_0 的位置即是基点．今后常称 $a\in S^Z$ 是(构形)空间中的点．

在 S^Z 中的任意两点 x 和 y 之间引入距离

$$d(x, y)=\sum_{i=-\infty}^{\infty}\delta(x_i, y_i)2^{-|i|},$$

其中当 $x_i=y_i$ 时 $\delta(x_i, y_i)=0$，当 $x_i\neq y_i$ 时 $\delta(x_i, y_i)=1$．然后，在 S^Z 中可以建立起开、闭、紧等等拓扑概念．这些对于今后的研究是必要的．

在 S^Z 上定义移位算子 σ 为

$$\sigma(x)_i=x_{i+1}, \qquad i\in Z.$$

现在定义与 σ 可交换的任何连续映射

$$F: S^Z\to S^Z$$

为元胞自动机．这里 F 与 σ 可交换的定义是 $F\circ\sigma=\sigma\circ F$，也就是对每个 $x\in S^Z$ 成立

$$F(\sigma(x))=\sigma(F(x)).$$

已经知道，这个定义与上一节中的定义是等价的．这也就是说，对于与 σ 可交换的连续映射 F，一定存在 $r>0$ 和局部映射 f: $S^{2r+1}\to S$，使得对每一个构形 $x=(\cdots x_{-1}x_0x_1\cdots)$ 和 $i\in Z$，成立

$$F(x)_i=f(x_{i-r}, \cdots, x_i, \cdots, x_{i+r}).$$

与局部映射 f 相对应，我们称 F 为元胞自动机的**全局映射**．它在今后讨论中是很有用的．

对于多维元胞自动机也有类似的结果．

取 S^Z 中所有构形在 F 作用下的象，就得到 S^Z 的象 $F(S^Z)$．显然有

$$S^Z\supseteq F(S^Z).$$

由此可见，对每个 $i\geqslant 0$，有

$$F^i(S^Z)\supseteq F^{i+1}(S^Z).$$

利用这样的单调包含关系，可以定义：

$$\Lambda(F)=\bigcap_{i=0}^{\infty} F^i(S^Z),$$

称为元胞自动机 F 的极限集. 由于 S^Z 在引进距离之后是紧集, F 是连续映射, 容易证明 $\Lambda(F)$ 一定不会是空集.

可以证明, 有

$$\lim_{i\to\infty} d(F^i(S^Z),\ \Lambda(F))=0,$$

其中 $d(\cdot,\ \cdot)$ 是两个集合之间的距离. 这个关系表明, 有意义的动力学行为在 $\Lambda(F)$ 之中.

利用定义可见

$$F(\Lambda(F))\subseteq\Lambda(F),$$

即 $\Lambda(F)$ 为 F 不变集. 实际上, $\Lambda(F)$ 是最大的 F 不变集.

§18.2 幂零型元胞自动机

考虑在极限集 $\Lambda(F)$ 中只含一个点(即构形)的最简单情况. 记 $\Lambda(F)=\{q\}$. 从 $\Lambda(F)$ 的 F 不变性可见有 $F(q)=q$. 另一方面, 从 $F\circ\sigma=\sigma\circ F$ 可以看出 $\Lambda(F)$ 也是 σ 不变集, 即从 $x\in\Lambda(F)$ 可推出 $\sigma(x)\in\Lambda(F)$. 由于这里 $\Lambda(F)$ 只含一个构形 q. 写出 $q=(\cdots q_{-1}q_0q_1\cdots)$, 可见每个 q_i 都相同. 这样的构形叫做空间平稳构形, 记为

$$q=(\cdots\bar{q}\bar{q}\bar{q}\cdots),\quad \bar{q}\in S.$$

我们称 $\Lambda(F)$ 只含一个构形的元胞自动机 F 为幂零型元胞自动机. 上面已经证明, 这个构形一定是空间平稳构形. 下面将要证明, 这类元胞自动机的动力学行为即是在 §17.4 中的第一类行为.

可以证明, $\Lambda(F)=\{q\}$ 的充分必要条件是, 从任何一个构形 $c\in S^Z$ 出发, 存在 n, 使 $F^n(c)=q$. 这里的 n 可以与 c 有关. 还可以证明, $\Lambda(F)=\{q\}$ 的充分必要条件是, 存在 N 使 $F^N(S^Z)=\{q\}$. 这等于是说, 对每个构形 $c\in S^Z$, 存在一个公共的 N, 使 $F^N(c)=q$ 对一切 c 同时成立.

用动力系统的语言来说，幂零型元胞自动机是最简单的情况。它有一个全局稳定的不动点$\{q\}$。所有轨道在有限时间内就直接落到这个不动点上。整个空间 S^Z 在有限时间内收缩到这个点上。不但如此，这个不动点本身还具有在空间上最为简单的平稳结构。

以上内容可参看[98]或其他文献。在论文[99]中证明，一维元胞自动机是否为幂零型乃是一个不可判定问题。这就是说，不存在一个算法，它关于任何一个一维元胞自动机，都能够对这个元胞自动机是否是幂零型的问题给出"是"或"否"的回答。

§ 18.3 $\Lambda(F)$ 为无限集的情况

这里叙述并证明关于极限集 $\Lambda(F)$的一个基本结果：如果 F 不是幂零型元胞自动机，那么 $\Lambda(F)$为无限集。换言之，$\Lambda(F)$不只含一个点时就一定含无限个点，没有其他可能性。利用 $\Lambda(F)$ 为移位不变集，只要找一个 $z\in\Lambda(F)$，使对于所有 $i\in Z$，$\sigma^i(z)$ 都各不相同，这样就已找到了 $\Lambda(F)$中的无限多个点。

从 σ 的定义可见，如果对 $i\neq j$，成立 $\sigma^i(z)=\sigma^j(z)$，那么 z 是双侧无限的周期符号序列。我们称这样的 z 为**空间周期构形**。在上一小节中所讲的空间平稳构形是它的特例，这时周期等于1。注意这里是关于 S^Z 中点的结构的概念，它与动力系统中的时间周期概念全然不同。

现设 x、$y\in\Lambda(F)$，且 $x\neq y$。如果在 x 与 y 中已经有一个不是空间周期构形，那么它就是上面所需要的 z。

现设 x 和 y 都是空间周期构形。不妨设 x 和 y 有相同的空间周期(否则取它们的公倍数)，记

$$x=(\cdots ww_{\vartriangle}w\cdots),\qquad y=(\cdots vv_{\vartriangle}v\cdots),$$

其中 w 和 v 为符号串，记号"$\vartriangle$"用于标出在它右方的第一个符号的位置即是 $i=0$ 的基点位置。从 $x\neq y$ 可见 $w\neq v$。但这里不排除 w 和 v 在循环移位下可以重合。如果这样，则 x 和 y 只相差某个空间移位。

定义在 S^Z 中的一个集合

$$N_w=\{(\cdots ww_{\triangle}\bar{w}\cdots)\mid \bar{w}\neq w\},$$

其中的构形在基点左方与 x 相同，而在基点右方长度为 $|w|$ 的子串与 w 不相同.

按照 S^Z 中的距离拓扑, N_w 是闭集. 同时可以看出, 在 N_w 中不含有任何空间周期构形. 于是, 只需要证明

$$N_w\cap \Lambda(F)\neq\varnothing.$$

由于 $\Lambda(F)=\bigcap_{i=0}^{\infty}F^i(S^Z)$，只要对每个 $i\geqslant 0$ 证明

$$N_w\cap F^i(S^Z)\neq\varnothing.$$

利用对每个 $i\geqslant 0$ 有 $x\in F^i(S^Z)$ 和 $y\in F^i(S^Z)$，可以将它们的原象适当拼接起来, 找到一个 $z\in F^i(S^Z)$, 它具有形状

$$z=(\cdots wws_1s_2\cdots s_nvv\cdots),$$

其中 $s_1s_2\cdots s_n$ 是某个符号串, 长度

$$n=2ir|w|\quad (=2ir|v|),$$

r 为局部映射 f 的邻域半径, $|w|=|v|$ 是 w 串和 v 串的长度, 即 x 与 y 的空间周期. 可看出 z 的某个移位 $\sigma^j(z)\in N_w$. 由于 $F^i(S^Z)$ 也是移位不变集, 因此也成立 $\sigma^j(z)\in F^i(S^Z)$. 证毕.

从今后的例子可知, 在 $\Lambda(F)$ 为无限集时，它可以是可列集，也可以是不可列集.

§18.4 周期点集合

记 $\mathrm{Per}(F)$ 为元胞自动机 F 的周期点集合. 在 $\mathrm{Per}(F)$ 为无限集时则考虑其闭包 $\pi(F)$.

容易证明，总有 $\mathrm{Per}(F)\neq\varnothing$，而且其中一定存在空间平稳的周期点. 注意，在未加说明时凡提到周期概念总是指时间上的周期行为.

任取一个空间平稳构形 c 为初始点. 考虑轨 $(c,\ F(c),\ \cdots,\ F^n(c),\ \cdots)$. 从局部映射可以看出, 每个 $F^i(c)(i\geqslant 1)$ 也是空间平

稳构形．因为 S 为有限符号集，故在 c、$F(c)$、…、$F^k(c)$ 中至少有两个相同，这里 k 是 S 所含的符号个数．这样就已经在 c、$F(c)$、…、$F^{k-1}(c)$ 中找到了周期点，而且其周期不超过 k．

如果从一个空间周期为 p 的初始构形 c 出发，那么情况也是类似的．这时每个 $F^i(c)(i\geq 0)$ 的空间周期都是 p，而长度为 p 的不同符号串只有 k^p 个．因此从 c 出发的轨必是周期轨或终极周期轨．这样也可找到周期不超过 k^p 的周期点．

但是以上推论并未表明元胞自动机一定有很多周期点．在 § 18.2 中的幂零型元胞自动机 F 的周期点集为

$$\mathrm{Per}(F)=\Lambda(F)=\{q\},$$

即只有唯一的一个周期为 1 的不动点．我们已经知道，q 的空间周期也是 1，即有 $\sigma(q)=q$．

更为突出的是在[98]中举了一个例子，它的周期点集 $\mathrm{Per}(F)$ 也只有一个不动点 q，同时 q 也是空间平稳构形．但这时的极限集 $\Lambda(F)$ 不等于 $\{q\}$．按照 § 18.3，这时的 $\Lambda(F)$ 为无限集．在[98]指出，这个元胞自动机具有复杂的动力学行为．

这里的情况与单峰映射大不一样．一个单峰映射如果只存在一个不动点，同时不存在任何其他周期轨的话，则它的动力学行为是极其简单的．在 § 13 定义的增长数 s 也是周期为 n 的周期点个数 P_n 的增长数．对于其他低维动力系统情况也是如此．因此在[98]中的例子是非平凡的．由于这个例子相当复杂，本书从略．

除了 $\Lambda(F)$ 和 $\mathrm{Per}(F)$（或 $\pi(F)$）这两个不变集之外，从动力系统的观点来看还可以研究 F 的非游荡集 $\Omega(F)$．关于这方面可参看[87]中的一般性介绍．明显有包含关系：

$$\pi(F)\subseteq\Omega(F)\subseteq\Lambda(F).$$

已经知道，其中每一个包含都可以是真包含．

§ 18.5 $\Lambda(F)$ 中点的逆向轨

设 $c\in\Lambda(F)$．从 $\Lambda(F)$ 为不变集可知，从 c 出发的轨完全在

$\Lambda(F)$中. 另一方面, 由于按 $\Lambda(F)$ 的定义, 它是单调下降的集合序列 $\{F^i(S^Z)\}_{i\geq 0}$ 的交, 因此对每个 $i\geq 0$, 有 $c\in F^i(S^Z)$. 这等于说对每个 $i\geq 0$, 存在点 c_{-i}, 使 $F^i(c_{-i})=c$.

可以将这个事实加强为: 在 $\Lambda(F)$ 中存在一个点序列 $\{c_{-i}\}_{i\geq 0}$, 使 $c_0=c$, 且对每个 $i\geq 1$ 有

$$F(c_{-i})=c_{-i+1}.$$

虽然映射

$$F:\quad S^Z\to S^Z$$

未必为可逆映射, 但可以将上述点序列 $\{c_{-i}\}_{i\geq 0}$ 看成点 c 的逆向轨(负半轨). 换句话说, 当 $c\in\Lambda(F)$ 时, 不仅从 c 出发的正半轨在 $\Lambda(F)$中, 而且也存在一条负半轨完全在 $\Lambda(F)$ 中. 这个结果有时是很有用的.

为此只要证明, 对 $c\in\Lambda(F)$, 存在一个点 $c_{-1}\in\Lambda(F)$, 使

$$F(c_{-1})=c.$$

首先, 从 $c\in F^n(S^Z)$对每个 $n\geq 0$ 成立, 可知有 $a_n\in S^Z$, 使 $F^n(a_n)=c$. 令

$$b_n=F^{n-1}(a_n),$$

那么 $F(b_n)=c$. 利用 S^Z 为紧集, 存在收敛子列 $\{b_{n_k}\}$. 记其极限为 b, 即有

$$\lim_{k\to\infty} b_{n_k}=b.$$

从 $F(b_{n_k})=c$ 中令 $k\to\infty$, 利用 F 的连续性, 就得到 $F(b)=c$. 以下证明这个 $b\in\Lambda(F)$, 因此它即是所要求的 c_{-1}.

对每个 k 有 $b_{n_k}\in F^{n_k-1}(S^Z)$. 利用 $\{F^i(S^Z)\}_{i\geq 0}$ 的单调下降性质和 $n_k\downarrow\infty$, 对每个 i, 当 k 充分大时就有

$$b_{n_k}\in F^i(S^Z).$$

令 $k\to\infty$, 可知极限点 $b\in F^i(S^Z)$. 由于 i 的任意性, 得到 $b\in\Lambda(F)$.

§19 元胞自动机中的正规语言

本节将证明，对每个 $i\geqslant 0$，取 $F^i(S^Z)$的所有有限子串而成的形式语言总是正规语言。讨论了几个初等元胞自动机的极限语言。本节材料主要取自[58]。

§19.1 $F(S^Z)$ 的复杂性

现在应用形式语言工具来研究 $F(S^Z)$的复杂性。由于每个构形是双侧无限的符号序列，采取与§6中相同的方法，即从每个构形取它的所有有限子串，将这样得到的子串全体作为研究的对象。将这个过程记为算子 $\mathscr{L}$，那么有 $\mathscr{L}(S^Z)=S^*$。相仿地，使用记号 $\mathscr{L}(F(S^Z))$、$\mathscr{L}(F^i(S^Z))$等等。此外，我们称 $\mathscr{L}(\Lambda(F))$为 F 的**极限集语言**。

现在证明 $\mathscr{L}(F(S^Z))$是正规语言。主要介绍 [58] 中的处理方法。关于这方面的其他研究方法，可参看[97]。在数学上的严格证明见[100]。

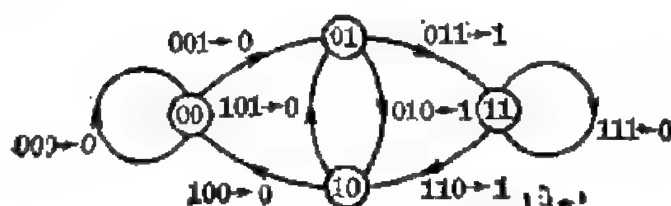


图 19-1

以图 17-1 中的 76 号初等元胞自动机为例来讨论。在图 19-1 上有四个结点，分别代表 00、01、10 和 11。从它们出发的弧分别反映 76 号规则中的内容。举例来说，从标以 01 的结点出发，根据规则中的 011→1，就有一条标以符号 1 的弧从 01 通向标以 11 的结点。又根据规则 010→1，又有一条标以 1 的弧通向标以 10 的结点。一般而言，从规则

$$a_i^{t+1}=f(a_{i-1}^t a_i^t a_{i+1}^t)$$

出发，从标以$a_{i-1}^t a_i^t$的结点有一条标以a_i^{t+1}的弧通向标以$a_i^t a_{i+1}^t$的结点.

在S^*中的任何一个符号串对应于图19-1中的一条路径. 例如,$00101\in S^*$对应于

$$00\to 01\to 10\to 01.$$

同时,其中三条弧对应的序列为010,即是在语言$\mathscr{L}(F(S^Z))$中的一个符号串. 于是,在语言$\mathscr{L}(F(S^Z))$中的每一个字对应于图中的一个路径.其中结点标号所成的序列在去掉重复出现符号后,即是$\mathscr{L}(S^Z)=S^*$中的字. 而弧的标号所成的序列即是$\mathscr{L}(F(S^Z))$中的字.

现在容易证明$\mathscr{L}(F(S^Z))$为正规语言. 仍以图19-1为例来说明这一点. 虽然这个有向图不是在本书§1中介绍的有限自动机，但只要以图19-1为基础，略加修改，就可以得到接受$\mathscr{L}(F(S^Z))$的一个有限自动机. 实际上，只要加入一个代表初态的结点,并用带ε转移的弧通向原有的每一个结点,再将每个结点定义为终止状态，就得到在§1.3中的带ε转移的非确定性有限自动机,它所接受的语言就是$\mathscr{L}(F(S^Z))$.

上述方法不仅证明了每一个$\mathscr{L}(F(S^Z))$是正规语言，而且可以推广到证明每一个$\mathscr{L}(F^i(S^Z))$也是正规语言，其中$i\geqslant 0$是任意非负整数.

如果将初始时刻的语言$\mathscr{L}(S^Z)=S^*$看成是没有任何结构的语言,那么从包含关系

$$S^Z\supseteq F(S^Z)\supseteq F^2(S^Z)\supseteq\cdots$$

可以看出也有

$$\mathscr{L}(S^Z)\supseteq\mathscr{L}(F(S^Z))\supseteq\mathscr{L}(F^2(S^Z))\supseteq\cdots.$$

因此可以设想,虽然对每个$i\geqslant 0$，$\mathscr{L}(F^i(S^Z))$是正规语言,但随着i的增加，$\mathscr{L}(F^i(S^Z))$可能会形成越来越复杂的结构. 又可以设想,如果按照关于正规语言的某种复杂性刻划，$\mathscr{L}(F^i(S^Z))$的复杂程度增长很快的话,那么极限集语言

$$\mathscr{L}(\Lambda(F))$$

很有可能是非正规语言．在[53]中用大量的计算机实验证实了对某些元胞自动机来说，以上设想是正确的，或者很可能是正确的．但在这方面还缺乏严格的数学论证．

§19.2 最小有限自动机

在 [53] 中对于正规语言 $\mathscr{L}(F^i(S^Z))(i\geqslant 0)$ 的复杂性刻划提出了几种方法：接受它的最小有限自动机的结点个数(即状态个数)，语言的熵和最短禁止字的长度．在本小节中我们讨论其中的第一种方法．

首先要指出，不论是 $\mathscr{L}(F^i(S^Z))(i\geqslant 0)$ 还是 $\mathscr{L}(\Lambda(F))$，都具有在 §7.4 中的两个基本性质，因此都属于在 §15.1 中所定义的 D 类语言．这里对于 §7.4 中的性质 2 还可以补充要求，对任何字 $x\in \mathrm{L}$，存在 $a\in S$，使 $ax\in \mathrm{L}$．实际上，这对于 $\mathscr{L}(\mathrm{KS})$ 也成立，只要取 $a=0$ 即可．

利用 §8.1 中的讨论，可见接受语言 $\mathscr{L}(F^i(S^Z))$ 的最小有限自动机只有一个非终止状态，它对应于语言的补集合．因此可以理解，在[53]中计算接受 $\mathscr{L}(F^i(S^Z))$ 的最小有限自动机的状态个数时一律不计入这个非终止状态，在状态转移图上也不画出它以及有关的弧．但在本书中仍采取与前面章节中统一的记号和作图规则．

按照[53]，采取以下两个步骤即可求出最小有限自动机．实际上，这与在[1]中介绍的一般方法完全相同．下面以图 19-1 为例作介绍．

第一步是消除自动机中的非确定性．将四个结点记为 u_0、u_1、u_2 和 u_3，其中下标值即是原有标号的二进制数值．然后取 $\{u_0, u_1, u_2, u_3\}$ 的所有子集，包括它自身和空集在内，共有 16 个子集．将每一个子集作为一个结点，然后从每个结点引出标以符号 0 和 1 的两条弧，它们通向哪一个结点可从图 19-1 决定．采用 §1.6 中

的右线性语法的记法，例如第一条规则为

$$\{u_0,\ u_1,\ u_2,\ u_3\} \to 0\{u_0,\ u_1,\ u_3\},$$

因为从四个结点出发沿着标以 0 的弧所到达的结点为 u_0、u_1 和 u_3，但不能到 u_2。

容易理解，$\{u_0,\ u_1,\ u_2,\ u_3\}$ 是我们要求的确定性有限自动机的初始状态，因为它包含了所有可能的四个状态. 从上述第一条规则继续做下去，可以得到

$$\{u_0,\ u_1,\ u_3\} \to 1\{u_2,\ u_3\},$$
$$\{u_2,\ u_3\} \to 0\{u_0,\ u_1,\ u_3\},$$
$$\{u_0,\ u_1,\ u_2,\ u_3\} \to 1\{u_2,\ u_3\},$$
$$\{u_0,\ u_1,\ u_3\} \to 0\{u_0,\ u_1,\ u_3\},$$
$$\{u_2,\ u_3\} \to 1\{u_2\},$$
$$\{u_2\} \to 0\{u_0,\ u_1\},$$
$$\{u_2\} \to 1\{\quad\} \qquad (\{\quad\} = \varnothing),$$
$$\{u_0,\ u_1\} \to 0\{u_0,\ u_1\},$$
$$\{u_0,\ u_1\} \to 1\{u_2,\ u_3\}.$$

以上十条规则已是所需的全部信息. 我们实际上只用了 16 个子集中的 6 个. 将$\{\quad\}(=\varnothing)$看成为非终止态，其他均为终止态，就可以得到图 19-2 中的有限自动机，其中 5 个黑圆点代表五个终止态，按从左到右的顺序为 $\{u_0,\ u_1,\ u_2,\ u_3\}$、$\{u_0,\ u_1,\ u_3\}$、$\{u_2,\ u_3\}$、$\{u_2\}$ 和 $\{u_0,\ u_1\}$.

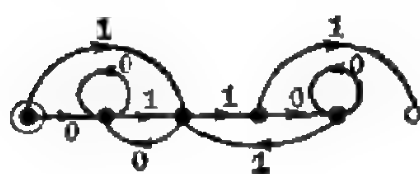


图 19-2

但这并不是最小有限自动机. 在[1]中介绍了这方面的一般算法. 下面以等价关系 R_L 和迈希尔-奈罗德定理为工具，说明对具体的问题如何在图上进行操作(参见 § 9.1 中的讨论).

利用在 § 9.1 中的命题:

$$\text{对每个 } a\in S,\ xaR_Lya \Rightarrow xR_Ly,$$

就可以将图 19-2 中的第一个结点和第二个结点合并. 实际上，从

这两个结点出发沿着标以 0 的弧都到达第二个结点，而沿着标以 1 的弧则都到达第三个结点. 因此知道，由这两个结点所代表的符号串都是 R_L 等价的，即可以合并. 这样就得到图 19-3. 实际上，这里的初态与从左方数起的第四个结点可以合并，用数学语言可以写成下列命题:

$$x1R_Ly1,\ x0R_Lx,\ y0R_Ly \Rightarrow xR_Ly.$$

这个合并之后得到图 19-4，它已是最小有限自动机. 这只要反复利用以下两个事实即可证明:

$$x\in L,\ y\notin L \Rightarrow xR_Ly \text{ 不成立}$$

$$xR_Ly \text{ 不成立},\ x'a=x,\ y'a=y \Rightarrow x'R_Ly' \text{ 不成立}.$$

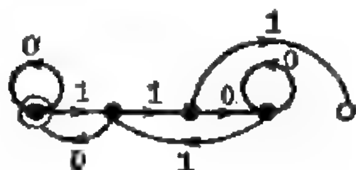


图 19-3

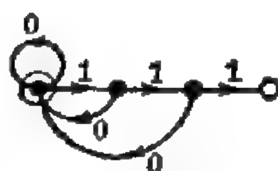


图 19-4

在[92]的附录(表 10)中对于 256 种初等元胞自动机($k=2$, $r=1$)，计算了 i 从 1 到 5 时的语言 $\mathscr{L}(F^i(S^Z))$ 的最小有限自动机的结点个数，并将这个指标称为正规语言的复杂性. 其中有一部分元胞自动机的这个指标在 i 增加时明显有指数增长速度. 这强烈地暗示它们的极限集语言 $\mathscr{L}(\Lambda(F))$ 为非正规语言，但至今尚无严格证明.

§ 19.3 76 号元胞自动机

由于在元胞自动机中的语言 $\mathscr{L}(F^i(S^Z))(i\geqslant 0)$ 和 $\mathscr{L}(\Lambda(F))$ 都是 D 类语言，我们可以将 § 15.1 中的禁止字概念用到这里来. 从图 19-4 直接可以看出，76 号元胞自动机的语言 $\mathscr{L}(F(S^Z))$ 只有唯一的一个禁止字 111.

另一方面，从图 17-1 可见，76 号元胞自动机的局部规则中，只要 $a_{i-1}^t a_i^t a_{i+1}^t$ 不是 111，就有 $a_i^{t+1}=a_i^t$. 因此，虽然 F 不是恒

等映射，但 F 作用于 $F(S^Z)$ 上则是恒等映射. 于是，就得到

$$F^i(S^Z)=F(S^Z), \qquad i\geqslant 1,$$

并且也有

$$\Lambda(F)=F(S^Z).$$

因此 76 号元胞自动机的极限集语言是正规语言，它的最小有限自动机即是图 19-4 所示的结果. 回顾图 17-2 中 76 号的计算机实验，可以看出情况确实是如此. 虽然在 $t=0$ 时刻的初始构形是随机选择的，但从 $t=1$ 开始，图形不会变化. 当然，各次实验所得的结果与初始构形有关.

从动力系统的角度来看，问题非常简单. 每一个不含子串 111 的构形都是不动点. 任何其他点出发的轨在一次迭代后就落到某一不动点上. 例如，在 § 12.4 和 § 14.4 中的帐棚映射 F_s 当参数值 $s=1$ 时就是如此. 76 号元胞自动机在沃尔夫勒姆的分类中属于第二类.

§ 19.4　128 号元胞自动机

这时的局部规则将 111 映到 1，而将其余映到 0. 在图 17-2 中画出了它的计算机实验结果，在很少几次迭代之后就得到完全由符号 0 组成的构形. 从表面上看似乎是属于沃尔夫勒姆分类中的第一类. 下面将要证明，如果将第一类行为等同于在 § 18.2 中的幂零型元胞自动机，那么 128 号并不属于这一类.

实际上，分别由符号 0 和 1 组成的空间平稳构形都是 F 的不动点. 因此在极限集 $\Lambda(F)$ 中至少含这两个点. 由 § 18.3 可见，$\Lambda(F)$ 为无限集. 我们将在下面对此作出全面的分析.

从动力系统的角度来看，可以认为 128 号元胞自动机有一个吸引不动点，即全 0 串，又有一个排斥不动点，即全 1 串. 极限集 $\Lambda(F)$ 由这两个不动点和连接它们的无限多条异宿轨道组成. 具体分析如下.

在图 19-5 中作出了接受 $\mathscr{L}(F(S^Z))$、$\mathscr{L}(F^2(S^Z))$ 和 $\mathscr{L}(\Lambda(F))$ 的最小有限自动机. 对于 $\mathscr{L}(F(S^Z))$，只有两个禁止

字：101 与 1001. 对于 $\mathscr{L}(F^2(S^Z))$，则有四个禁止字：10^n1，$1\leqslant n\leqslant 4$. 用归纳法容易证明，对于 $\mathscr{L}(F^i(S^Z))$，有 $2i$ 个禁止字：10^n1，$1\leqslant n\leqslant 2i$. 它们都是有限补语言.

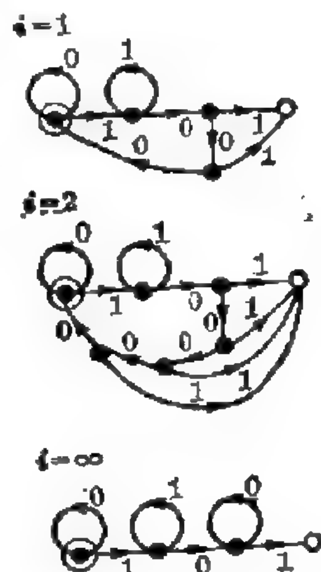


图 19-5

接受语言 $\mathscr{L}(\Lambda(F))$ 的最小有限自动机只有四个状态，它的复杂性在于这时有无限多个禁止字. 如记 $L=\mathscr{L}(\Lambda(F))$，则按 § 15.1 中的记号，禁止字集

$$L'=\{10^n1\mid n\geqslant 1\}.$$

语言 $\mathscr{L}(\Lambda(F))$为无限补语言.

从图 19-5 直接可以写出 $\mathscr{L}(\Lambda(F))$ 的正规表达式为

$$\mathscr{L}(\Lambda(F))=0^*1^*0^*.$$

这表明，在极限集 $\Lambda(F)$中的构形，除了全 0 串和全 1 串之外，其中出现的符号 1 必须连接成一个串. 容易理解，这些都是游荡点. 这时

$$\mathrm{Per}(F)=\Omega(F)=\{\bar{0},\bar{1}\}\subsetneq\Lambda(F).$$

容易直接验证在 § 18.5 的命题. 即对每个 $c\in\Lambda(F)$，不仅由 c 出发的正半轨在 $\Lambda(F)$中，同时存在按 § 18.5 意义下的一条负半轨也在 $\Lambda(F)$中.

最后，容易说明为什么在实验中，对 128 号元胞自动机来说，从随机的初始构形出发总能很快就收敛到全零串. 实际上，如在 $F^i(c)$中含有一个符号 1，则在 c 中至少要有 $2i+1$ 个 1. 对于用 N 个元胞和周期边界条件的计算机实验来说，如果 $i>N/2$，则等于要求 N 个元胞的初态全是 1. 由于初始构形是随机选取的，因此取到这种初始构形的概率为 2^{-N}. 当 N 很大时，实际上是不可能出现的.

§ 19.5 90 号元胞自动机

从计算机实验中明显可见它的动力学行为属于第三类. 但是

从 90 号元胞自动机的局部规则为:

$$111\to0,\quad 110\to1,\quad 101\to0,\quad 100\to1,$$
$$011\to1,\quad 010\to0,\quad 001\to1,\quad 000\to0,$$

可以看出有

$$F(S^Z)=S^Z,$$

即 F 为将 S^Z 映到自身上的满射. 实际上, 一般而言只要在局部映射 $a_i^{t+1}=f(a_{i-1}^t,\ a_i^t,\ a_{i+1}^t)$ 中, 如 f 关于 a_{i+1}^t 为单射, 或关于 a_{i-1}^t 为单射, 就成立 $F(S^Z)=S^Z$. 从图 19-1 来说明这一点是很直观的. 这时从每个结点出发(或进入每个结点)的两条弧分别标以符号 0 和 1, 从而就保证了在 $F(S^Z)$ 中出现任何构形.

这表明研究 $\Lambda(F)$ 以及语言 $\mathscr{L}(\Lambda(F))$ 的方法所具有的局限性. $\Lambda(F)$ 是 F 的最大不变集, 它本身不一定能够反映 F 在 $\Lambda(F)$ 上的复杂动力学行为. 在 F 为满射时这是非常明显的事实. 打个比方来说, 在二次方映射为满射的情况, 即 $f(x)=4x(1-x)$, 这时的最大不变集为区间 $I=[0, 1]$ 本身. 仅仅这一点不能说明什么问题.

规则 90 号是可加元胞自动机的一个例子, 这时即为

$$f(a_{i-1},\ a_i,\ a_{i+1})=a_{i-1}+a_{i+1} \bmod 2.$$

这方面已有很多研究. 在周期边界条件下可以用代数方法求解(见[101]. 又已证明, 在初始构形只含有限个符号 1 的条件下, 90 号元胞自动机的每一个元胞的状态变化序列 $\{a_i^n\}_{n>0}$, 除了一种平凡情况外, 一定是非周期的. 这揭示了 90 号的混沌行为的一个方面(参看[102]). 在本书 § 21.3 中将计算 90 号元胞自动机的拓扑熵, 说明它与二次方映射中的满射情况有一个相似之处, 即拓扑熵都达到极大.

§ 19.6 18 号与 22 号元胞自动机

在图 17-2 中画出了这两个元胞自动机的实验结果. 图 19-6 是当 F 为 18 号元胞自动机时, 接受语言 $\mathscr{L}(F(S^Z))$ 的最小有限

自动机．它的最短禁止字是 111．可以看出，这个 $\mathscr{L}(F(S^z))$是一个无限补正规语言．它的禁止字集合可用正规表达式表示为

$$L''=11(0^+10^+1)^*1,$$

其中 $0^+=00^*$.

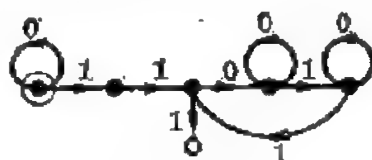


图 19-6

对于 18 号元胞自动机来说，用接受 $\mathscr{L}(F^i(S^z))$ 的最小有限自动机的结点个数来刻划的正规语言复杂性如下：$i=1$ 时为 6(即图 19-6)，$i=2$ 时为 48，$i=3$ 时为 144，$i=4$ 时估计已大于 20000．因此可以推测，语言 $\mathscr{L}(\Lambda(F))$不是正规语言．但至今尚无证明．

可以建立在 18 号与 90 号元胞自动机之间的一个联系．如果只考虑由 00 和 01 组成的构形，并且令 $00\to0, 01\to1$，那么 18 号的两次迭代恰好模拟了 90 号的一次迭代．从 90 号具有性质

$$F(S^z)=S^z$$

可以推知，在 18 号元胞自动机的极限集 $\Lambda(F)$中，至少含有所有不含子串 11 的构形全体．

关于 22 号元胞自动机也有类似情况．当 F 为 22 号时，语言 $\mathscr{L}(F^i(S^z))$ 的正规语言复杂性在 $i=1$ 时已为 16，也就是说与图 19-2 对应的那个确定性有限自动机已含 15 个终止状态，而且它已是最小有限自动机．在 $i=2$ 时为 281，在 $i=3$ 时为 4507，在 $i=4$ 时估计已大于 20000.

同样，如令 $0000\to0$，$0001\to1$，则 22 号的四次迭代恰好模拟 90 号的一次迭代．

18 号和 22 号元胞自动机的材料很多，除了[92]、[93]，还可参看[41]等，但尚未得到完整的结果．

§20 元胞自动机中的非正规语言

虽然关于初等元胞自动机中可能出现非正规语言的猜测没有

得到数学上的证明或否定，但在元胞自动机中的非正规语言方面还是取得了不少成果. 本节材料主要取自 [103]、[104].

§20.1 四类行为的出现频率

上面已经提到，有大量的计算机实验使我们有理由猜测，很多元胞自动机的极限集语言为非正规语言. 这也就是说，它们的极限集具有较高程度的复杂性. 表 20.1 列出了在合法完全的元胞自动机中，四类不同的动力学行为所占的大致比例(此表取自 [92]). 这里关于合法和完全的定义见 §17.2. 如 §17.4 所说，在 $k=2$ 和 $r=1$ 的初等元胞自动机中不出现第四类动力学行为. 关于分类问题可看 [96] 以及在论文集 [93] 中的其他论文.

表 20.1

	$k=2$ $r=1$	$k=2$ $r=2$	$k=2$ $r=3$	$k=3$ $r=1$
1	0.50	0.25	0.09	0.12
2	0.25	0.16	0.11	0.19
3	0.25	0.53	0.73	0.60
4	0	0.06	0.06	0.07

从 §19.5 已经看到，如果 $\Lambda(F)$ 很简单，则 F 仍可以有复杂的动力学行为. 但当 $\mathscr{L}(\Lambda(F))$ 为一个比较复杂的语言时，则从 $\Lambda(F)$ 的定义

$$\Lambda(F)=\bigcap_{t=0}^{\infty}F^t(S^Z)$$

和性质

$$\lim_{t\to\infty}d(F^t(S^Z),\ \Lambda(F))=0,$$

可以推测相应的元胞自动机 F 会有复杂的动力学行为.

直到目前为止，据作者所知还没有对于表 20.1 中的任何一个具有第三或第四类动力学行为的元胞自动机，作出 $\mathscr{L}(\Lambda(F))$ 为

非正规语言的数学证明.

下面介绍赫德(L. P. Hurd)的一些结果.

§20.2 $\mathscr{L}(A(F))$为上下文无关语言的例子

以一个上下文无关语言$\{0^n10^n|n\geq 0\}$为基础(见§1.7与§3.2), 可以构造出一个$k=6$和$r=2$的元胞自动机F, 使$\mathscr{L}(A(F))$为上下文无关语言, 但不是正规语言.

取$k=6$的符号集

$$S=\{0, W, r, R, l, L\},$$

又取半径$r=2$, 并给定以下局部规则

1. $\cdot\cdot W\cdot\cdot \to W$,
2. $\cdot\cdot RWL \to l$,
3. $RWL\cdot\cdot \to r$,
4. $\cdot rxy\cdot \to r(x\neq W,\ x、y\neq l、L)$,
5. $\cdot Rxy\cdot \to R(x\neq W,\ x、y\neq l、L)$,
6. $\cdot xyl\cdot \to l(y\neq W,\ x、y\neq r、R)$,
7. $\cdot xyL\cdot \to L(y\neq W,\ x、y\neq r、R)$,
8. 其余情况$\to 0$.

在以上规则中, 记号"·"代表S中的任何符号. 此外, 如果同时可以应用一条以上的规则, 则按上述顺序在先者为优先.

容易解释上述定义的物理意义. 将S中除了符号0之外的五个符号设想为在直线上的五种粒子, 它们的位置均在整数点处. 符号0代表不被上述粒子占有的位置. W代表不移动的粒子. L和l代表向左移动的两种粒子, R和r代表向右移动的两种粒子. 它们的速度均为每个单位时间移动一个单位距离, 即相邻整数点之间的距离. 规则2~8表示可移动粒子相撞后消失(湮灭), 但有一个特殊情况, 即当R和L粒子同时(从相反方向)与W粒子相撞时, 则R粒子变为l粒子, 而L粒子变为r粒子. 图20-1是对这些规则的一个说明. 其中空格均由符号0占有, 不再画出.

现在将上述元胞自动机(的全局映射)记为 F. 可以证明

$$\mathscr{L}(\Lambda(F))=K_1\cup K_2,$$

其中 K_1 和 K_2 的表达式为:

$$K_1=(r+R+0)^*(W+0)^*(l+L+0)^*,$$

$$K_2=\{(r+R+0)^*l0^iW0^ir(l+L+0)^*\mid i\geqslant 0\}.$$

实际上,任取 $z\in K_1$ (或 K_2), 在两侧补充符号 0 使成为 S^Z 中的一个构形,容易看出,对每个 $i>0$, 这样的构形有第 i 次原象, 即属于 $F^i(S^Z)$中,这样就证明了

$$K_1\subseteq\mathscr{L}(\Lambda(F)),\qquad K_2\subseteq\mathscr{L}(\Lambda(F)).$$

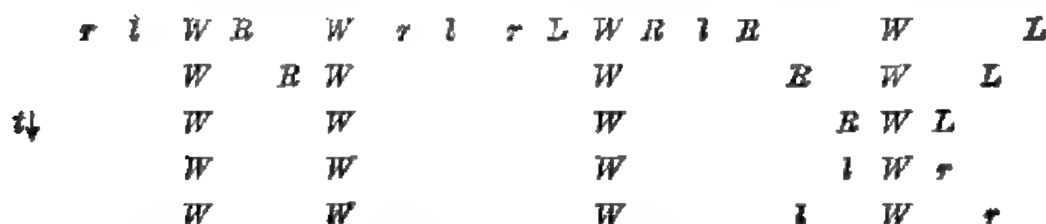


图 20-1

反之,对于 $\Lambda(F)$中的任一构形, 如含有一个以上的符号 W, 则从 § 18.5 可知,在 W 与 W 之间只能由符号 0 来填满. 也就是说,在不动粒子 W 之间的任何可移动粒子不可能无限期地存活下去. 对于(可能存在的)最左的 W 粒子,那么在它的左方可以有 r 和 R 粒子. 同样对于(可能存在的)最右的 W 粒子, 在它的右方可以有 l 和 L 粒子. 以上情况的唯一可能例外是在 K_2 中反映的情况,即在 $\Lambda(F)$的构形中只有一个 W 粒子,在它的左方有一个 l 粒子,右方有一个 r 粒子,且与 W 粒子距离相等. 同样利用 § 18.5 可知这样的构形在 $\Lambda(F)$中是存在的. 它代表了在前述特殊碰撞之后的情况. 这样我们就知道在 $\Lambda(F)$ 中没有其他构形. 从而得到 $\mathscr{L}(\Lambda(F))$为 K_1 和 K_2 的并的结论.

由于 K_1 是用正规表达式写出的,因此是正规语言. 从 K_2 的表达式是一个上下文无关语言 $\{0^iW0^i\mid i\geqslant 0\}$ 与正规语言连接而成,可知 K_2, 以及 $\mathscr{L}(\Lambda(F))$本身,都是上下文无关语言.

最后证明，$\mathscr{L}(A(F))$不是正规语言．否则，

$$\mathscr{L}(A(F))\cap l0^*W0^*r=\{l0^iW0^ir\mid i\geqslant 0\}$$

也成为正规语言了．证毕．

§20.3 $\mathscr{L}(A(F))$为上下文有关语言的例子

利用类似的方法，作出了 $k=6$ 和 $r=4$ 的一个元胞自动机 F，它的极限集语言 $\mathscr{L}(A(F))$为上下文有关语言，但不是上下文无关语言．以下简要地叙述与上一个例子的不同之处．

已知 $\{a^ib^ic^i\mid i\geqslant 1\}$ 为上下文有关语言，但不是上下文无关语言(参见§3.3, §3.6 和§4.2)．

取相同的符号集

$$S=\{0,\ W,\ r,\ R,\ l,\ L\},$$

但这时 R 和 L 的速度加倍(快粒子)，而 r 与 l 仍与上例相同．对于特殊情况的碰撞，修改为

$$rrWll\to LlWrR,$$

其他情况的碰撞与上例相同．这时令 $r=4$，具体的局部映射为

1. $\cdot\cdot\cdot\cdot W\cdot\cdot\cdot\cdot\to W,$
2. $rrWll\cdot\cdot\cdot\cdot\to R,$
3. $\cdot rrWll\cdot\cdot\cdot\to r,$
4. $\cdot\cdot\cdot\cdot rrWll\to L,$
5. $\cdot\cdot\cdot rrWll\cdot\to l,$
6. $\cdot\cdot arbcd\cdot\cdot\to r(a\neq R,\ b\neq W、l、L,\ c\neq l、L,\ d\neq L),$
7. $\cdot\cdot Rabcd\cdot\cdot\to R(a\neq r、W、l、L,\ b\neq W、l、L,$
   $c\neq l、L,\ d\neq L),$
8. $\cdot\cdot abcld\cdot\cdot\to l(d\neq L,\ c\neq W、r、R,\ b\neq r、R,\ a\neq R),$
9. $\cdot\cdot abcdL\cdot\cdot\to L(d\neq l、W、r、R,\ c\neq W、r、R,$
   $b\neq r、R,\ a\neq R),$
10. 其余情况 $\to 0$．

与上例一样，按上述顺序规定优先级．在图 20-2 举了一个例子，

用以说明规则的使用方式. 现在记这个元胞自动机为 F, 则可以证明

$$\mathscr{L}(\Lambda(F))=S_1\cup S_2,$$

$$S_1=(R+0)^*(r+0)^*(W+0)^*(l+0)^*(L+0)^*,$$

$$S_2=\{(R+0)^*(r+0)^*L0^il0^iW0^ir0^iR(l+0)^*(L+0)^*\mid i\geqslant 0\}.$$

然后证明 $\mathscr{L}(\Lambda(F))$是上下文有关语言, 但不是上下文无关语言. 具体证明过程这里从略.

r　　L　l W　r r　W　l l　r　R

$t\downarrow$　r L　l　W　r r W l l　r

l　W　L l W r R　r

l　W L　l　W　r　R

图 20-2

§ 20.4 关于复杂性的一些理论结果

在研究 $\Lambda(F)$的复杂性时, 已经知道有一个普遍性的限制.

设 F 为元胞自动机, S 为符号集. 可以证明, 极限集语言 $\mathscr{L}(\Lambda(F))$在 S^* 中的补(即 $S^*-\mathscr{L}(\Lambda(F))$)一定是递归可枚举语言, 即在乔姆斯基层次体系中最高一层的语言(见 § 2.4、§ 2.5 和 § 3.1). 这即是图灵机能够识别和枚举的语言.

这个结论的意义在于, 如果将大量存在的非递归可枚举语言看成是复杂程度不可计算的情况, 那么对元胞自动机来说, 至少 $S^*-\mathscr{L}(\Lambda(F))$的复杂程度仍属于可计算的范围中.

证明是容易的. 设有

$$z\in S^*-\mathscr{L}(\Lambda(F)),$$

那么从定义有

$$\mathscr{L}(\Lambda(F))=\bigcap_{i=0}^{\infty}\mathscr{L}(F^i(S^Z)),$$

因此, 至少存在一个自然数 n, 使

$$z \notin \mathscr{L}(F^n(S^Z)).$$

现设元胞自动机的邻域半径为 r. 我们可以说明，对任何自然数 $m>0$，一个长度为 m 的给定符号串 z 是否属于 $\mathscr{L}(F^n(S^Z))$，这是一个只需要有限步计算即可解决的问题. 实际上，利用 F 与 σ 可交换，考虑所有长度为 $m+2nr$ 的符号串，它们共有 k^{m+2nr} 个，逐个检查它们在 F^n 映射下的象是否等于 z 就可以解决上述问题.

现在可以设计一个图灵机来做上述工作. 从 $i=1$ 开始做起，在发现 $z\in\mathscr{L}(F^i(S^Z))$时，对 $i+1$ 做同样的计算，直到发现 $z\notin\mathscr{L}(F^n(S^Z))$为止. 这时图灵机接受 z 而停机. 这样的图灵机即是接受语言 $S^*-\mathscr{L}(\Lambda(F))$的自动机，因此这是一个递归可枚举语言.

但对于 $z\in\mathscr{L}(\Lambda(F))$，则上述图灵机将无休止地计算下去. 因此上述命题并没有对语言 $\mathscr{L}(\Lambda(F))$带来什么限制. 实际上，在[104]中已经用抽象的方法证明，存在一个元胞自动机 F，它的极限集语言 $\mathscr{L}(\Lambda(F))$ 不是递归可枚举语言. 与本书 § 12.4 比较，我们看到在单峰映射和在元胞自动机中，都存在不可计算的复杂性. 当然，不可能给出这方面的具体例子，因为这直接违反了在 § 2.4 中的丘奇-图灵论题.

此外，在元胞自动机中存在着大量的不可判定问题，其中包括在 § 18.2 中的幂零问题以及在上面的 $z\in\mathscr{L}(\Lambda(F))$的判定问题. 卡里(J. Kari)已经证明，关于元胞自动机的任何一个非平凡命题都是不可判定的(参见[99]、[98]). 这里的不可判定问题的含义是，不存在一个有效算法，它可以对一切元胞自动机在这个问题上作出“是”或“否”的回答. 当然，这完全不排除可以对一个或一类元胞自动机在相同问题上作出彻底的分析.

关于 $\pi(F)(=\overline{\mathrm{Per}(F)})$ 已经证明，任何 $\mathscr{L}(\pi(F))$ 一定是递归可枚举语言. 但在[104]中已经举出抽象的例子，它的 $\mathscr{L}(\pi(F))$ 不是递归语言(参见 § 2.5 或 [1]).

§21 空间熵与时间熵

这里的时间熵即是元胞自动机的拓扑熵，它是在[98]中建立的.

§21.1 两种不同的熵

在[53]中对语言 $\mathscr{L}(F^i(S^Z))$ 计算了它的熵. 这里熵的定义与本书§13.1完全相同，其中的计算方法则是在§13.4中的伴随矩阵方法.

在§13.6已经指出，对语言 $\mathscr{L}(\mathrm{KS})$ 定义的熵即是单峰映射的拓扑熵，它反映了动力系统的轨道的多样性. 在这个意义上，我们说在§13.1中定义的熵具有动力学意义.

但在[53]中所计算的语言 $\mathscr{L}(F^i(S^Z))$ 的熵则全然不同，它反映了在固定时刻 $t=i$ 时映象 $F^i(S^Z)$ 中空间构形的多样性，并不具有时间方向上的动力学意义. 因此在[53]中称这样的熵为空间熵是合适的.

从包含关系

$$S^Z \supseteq F(S^Z) \supseteq F^2(S^Z) \supseteq \cdots$$

可以看出，随着 i 的增加，语言 $\mathscr{L}(F^i(S^Z))$ 的熵是单调不增序列. 可以认为，这反映了元胞自动机随着时间演化有可能表现出一定的自组织能力. 因此研究这样的序列也许可获得一定的信息. 但是这里与 $\Lambda(F)$ 的研究相仿，在遇到 F 为满射等情况时就不可能得到什么.

在[53]中已经提出了时间熵的概念，指出要与上述空间熵相区别. 赫德在[98]中第一次对于时间熵（即拓扑熵）给出了计算公式，研究了它的基本性质，这就是本节要介绍的主要内容.

§ 21.2 元胞自动机的拓扑熵计算

将元胞自动机 F 作为动力系统来研究，即将映射 $F: S^Z \to S^Z$ 迭代而成动力系统时，则在 [53] 中提出的时间熵即是该动力系统的拓扑熵. 在[98]中给出了拓扑熵的计算公式：

$$h(F)=\lim_{w\to\infty}\left(\lim_{t\to\infty}\frac{\log R(w,\ t)}{t}\right),$$

其中 $R(w, t)$是元胞自动机 F 在空间宽度为 w，时间高度为 t 的矩形上所有可能图式的个数.

根据在 § 13.6 中对拓扑熵的介绍，容易理解上述计算公式的正确性及其含义.

设符号集 S 中符号个数为 k. 利用元胞自动机的平移不变性，即 F 与 σ 可交换性，只要考虑元胞位置 i 满足 $0\leqslant i<w$ 的一段. 这时可能出现的全部(长为 w 的)符号串共有 k^w 个. 将 S^Z 中所有构形按照在这 w 个元胞位置上的状态来分类，即将在这段上相同的构形作为一个类，就可以将 S^Z 划分成 k^w 个子集合. 按照在 S^Z 中引入的距离拓扑，它们都是开集，而当 $w\to\infty$ 时每一个开集的直径趋于 0. 按照 § 13.6，只要对这个划分(序列)来计算拓扑熵就可以了.

如记上述划分为 $\widetilde{w}$，则按 § 13.6 中的公式，就有

$$h(F)=\lim_{w\to\infty}h(F,\ \widetilde{w})=\lim_{w\to\infty}\left(\lim_{t\to\infty}\frac{1}{t}\ H\left(\bigvee_{i=0}^{t-1}F^{-i}(\widetilde{w})\right)\right).$$

考虑划分 $\bigvee_{i=0}^{t-1}F^{-i}(\widetilde{w})$中的每一个子集合. 如果 c 属于这个加细划分中的某个子集，那么 c、$F(c)$、…、$F^{t-1}(c)$ 在 $0\leqslant i<w$ 这一段上的符号串都是确定的. 因此，这个加细划分的子集个数就是上面给出的 $R(w, t)$，即有

$$N\left(\bigvee_{i=0}^{t-1}F^{-i}(\widetilde{w})\right)=R(w,\ t).$$

因此就得到 $$H\left(\bigvee_{i=0}^{t-1}F^{-i}(\widetilde{w})\right)=\log R(w,\ t),$$

计算公式已经证明成立.

从这个公式容易给出 $h(F)$的上界估计.

设 F 的邻域半径为 r, 那么 $R(w,\ t)$ 不会超过在 $t=0$ 时刻的长度为 $w+2r(t-1)$的元胞状态的可能符号串个数:

$$R(w,\ t)\leqslant k^{w+2r(t-1)}.$$

代入公式中, 就得到拓扑熵的上界:

$$h(F)\leqslant 2r\log k.$$

§ 21.3 举例

[例 1] 设 $S=\{0,\ 1,\ \cdots,\ k-1\}$, 那么移位算子 σ 本身也是一个元胞自动机. 对于 $k=2$ 和 $r=1$, 这就是 170 号初等元胞自动机. 这时易见有

$$R(w,\ t)=k^{w+t-1}.$$

代入公式, 得到 σ 的拓扑熵为

$$h(\sigma)=\log k.$$

这是遍历理论中的一个典型例子(参见[87]).

[例 2] 在上一小节中给出的拓扑熵的上界是可以达到的. 仍设 $S=\{0,\ 1,\ \cdots,\ k-1\}$. 定义局部映射为具有可加性质的

$$f(a_{i-r},\ \cdots,\ a_i,\ \cdots,\ a_{i+r})=\sum_{j=-r}^{r}a_{i+j}\ \mathrm{mod}\ k.$$

可以证明这时的

$$R(w,\ t)=k^{w+2r(t-1)},$$

即已达到最大可能的多样性, 从而其拓扑熵达到上界 $2r\log k$.

现在证明关于 $R(w,\ t)$的上述公式.

用反证法. 如果此公式不成立, 则表明在 $t=0$ 时刻有两个长为 $w+2r(t-1)$的不同符号串, 它们在 $w\times r$ 的矩形中生成完全相同的图式. 由于上述元胞自动机具有可加性, 在相减后就得到在 $t=0$时刻的一个不全为 0 的符号串, 它在 $w\times r$ 矩形中生成完全为 0 的图式. 在图 21-1 中作出了示意图. 从熵的计算公式可以看

出，$w\to\infty$为外层极限，因此不妨已取$w>2r$．从图 21-1 可以看出，在$t=0$时刻的长为$w+2r(t-1)$的符号串中，在中央的长为w的子串已全由符号 0 组成，因为它是$w\times r$矩形中的一部分．由此出发，利用$w>2r$，即可证明$t=0$时刻的长为$w+2r(t-1)$的符号串完全由符号 0 组成．这与上述假定相矛盾．

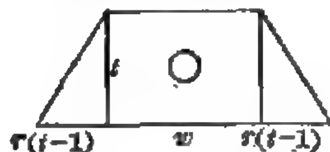


图 21-1

［例 3］ 例 2 在初等元胞自动机中的特例即是 150 号：

$$111\to1,\quad 110\to0,\quad 101\to0,\quad 100\to1,$$

$$011\to0,\quad 010\to1,\quad 001\to1,\quad 000\to0.$$

从§19.5 的讨论可见，这时的元胞自动机F也是从S^Z到自身上的满射．即有

$$F(S^Z)=S^Z,\qquad \Lambda(F)=S^Z.$$

从计算机实验(见[92]的表 2)可知 150 号元胞自动机的动力学行为明显属于第三类，即混沌行为．从极限集$\Lambda(F)$不能反映出这一点，但拓扑熵达到极大则正确地反映了动力系统的混沌特性．可以将这种情况与单峰映射中的满射情况相比拟．

［例 4］ 对于 90 号元胞自动机可以作类似的讨论．容易证明，这时的熵也达到了极大值 2$(r=1,\ k=2)$．在§19.5 我们已经知道，用极限集方法来讨论 90 号元胞自动机是不成功的．因此，可以认为在[98]中给出的拓扑熵计算公式，从轨道多样性的角度对于 90 号和 150 号元胞自动机的动力学行为给出了一个令人满意的刻划．

［例 5］ 对于 204 号元胞自动机，局部映射规则为

$$111\to1,\quad 110\to1,\quad 101\to0,\quad 100\to0,$$

$$011\to1,\quad 010\to1,\quad 001\to0,\quad 000\to0.$$

实际上，这个元胞自动机是在S^Z上的恒等映射．当然，它也是满射，有$F(S^Z)=S^Z$和$\Lambda(F)=S^Z$成立．因此从计算空间熵来看，

与上面的150号和90号元胞自动机毫无区别，得到的值都是1。它反映了符号串的多样性已达到最大. 但这丝毫不反映204号在动力学行为上的简单性.

用拓扑熵计算公式，容易求出这时的

$$R(w,\ t)=2^{w},$$

代入公式即得到 $h(F)=0$. 它正确地反映了204号在动力学行为方面的特征.

除去以上例子外，还容易证明，在§18.2研究过的幂零型元胞自动机的拓扑熵一定是0.

还可以用拓扑熵方法讨论在§19研究过的76号、128号、18号和22号元胞自动机，证明前两个的拓扑熵为0，而后两个的拓扑熵都是大于0的。

§21.4 理论上的限制

如上所述，拓扑熵可以成功地应用于某些具体的元胞自动机的研究中. 但目前缺少关于 $R(w,\ t)$ 的计算方法，因此就限制了拓扑熵的应用. 在这方面了解一下理论上的限制是有必要的.

在[98]中利用计算理论得到了下面列出的三个否定性结论:

1. 不存在一种算法，它可以对每一个元胞自动机判定其拓扑熵是否为0.

2. 对于任意给定的正数 s，不存在一种算法，它可以对每一个元胞自动机判定其拓扑熵是否大于 s.

3. 对于任意给定的正数 s，不存在一种算法，它可以对每一个元胞自动机计算其拓扑熵的近似值，使误差不超过 s。

当然，这丝毫不能说明对很多元胞自动机来说，我们不能求出它们的拓扑熵. 尽管有以上的限制，拓扑熵仍然可能发展成为研究元胞自动机的重要工具. 它们只是告诉我们，要设计出对一切元胞自动机都适用的拓扑熵算法是不可能的。

在[98]中还证明，存在使

$$\mathrm{Per}(F)=\pi(F)=\{q\}\subsetneq\Lambda(F)$$

成立的例子．这在§18.4中已经提到．在[98]中还指出，这类例子中的拓扑熵可以具有任意大的数值，这表明相应的动力学行为是很复杂的．

第 7 章

单个序列的复杂性

本章研究对象为单个序列的复杂性分析. 这与本书前面各章都不相同. 也可以用本章的方法来研究动力系统, 这时, 不同的轨道可以有不同的复杂性.

已经清楚, 关于单个序列的复杂性概念与随机性概念是不能分开的. 这里要指出, 后者与概率论中的随机概念是不一样的. 问题的特点是只研究给定的一个序列, 不问其来源如何.

本章的前两节介绍柯尔莫哥洛夫(А. Н. Колмогоров)复杂性. 它在单个序列的复杂性研究中具有根本的意义, 但在具体应用中则不如后两节的两种刻划方法来得方便.

如何分析一个给定序列的复杂性是一个有实际意义的重要问题. 本章只是一个很肤浅的介绍, 其中有不少内容本身也不很成熟, 尚处于发展之中, 希望对读者能起到一点抛砖引玉的作用.

§22 柯尔莫哥洛夫复杂性

这一节主要介绍在单个序列的复杂性刻划中的困难所在, 提出柯尔莫哥洛夫复杂性的定义. 主要材料取自[105].

§22.1 单个符号序列的复杂性

符号序列出现在极为众多的各个科学技术领域中, 例如通信、密码、遗传工程等等. 经常出现的问题是对于一个给定的符号序列, 如何分析它的复杂性. 从直观上可以知道, 某些序列是简单的, 例如

$$0000000000000000, \qquad 1010101010101010,$$

当然一点不复杂．另一方面，对于

$$0010101110111010, \qquad 0110001001111011,$$

就不容易说它们是否复杂了．在平时我们也往往使用随机性这个词来代替复杂性．例如，说前两个序列不是随机的，而后两个序列则似乎是随机的．

从一开始就应当指出，上述说法与概率统计理论中的随机概念是完全不一样的，后者是有严格定义的数学概念．因此，像上面那样用随机性代替复杂性的说法并没有给我们带来任何东西．

下面主要讨论有限长度的符号序列．与本节其余部分一样，用 S 表示一个有限符号集，用 S^* 表示取自 S 的有限序列全体．与本书其余部分不同的是，我们在这里研究 S^* 中一个给定序列的复杂性刻划问题．

对于这样的问题，乔姆斯基的层次体系不能提供什么结果．事实上，对于只含一个符号串的语言 $L=\{z\}\subseteq S^*$，只能说它是正规语言(或者有限语言)．容易知道，接受 L 的最小有限自动机具有平凡的结构，它的状态个数只与 z 的长度有关，而与 z 本身是否具有某种复杂性无关．因此在这里需要建立完全不同的理论．

首先指出，不可能有任何一个合理的方法，它能将符号串全体简单地划分成复杂和简单的两类．用反证法．如果存在这样的方法，那么在本节一开始举出的两个序列当然属于简单序列．我们不知道后面两个序列按照这个方法是划分在简单类还是复杂类中，但总应该存在某一个序列，它在这种划分方法下是复杂序列．此外，在一个复杂序列中任意改变某一个符号，不应当使这个复杂序列一下子就变成为简单序列，这也是一个合理的要求．

由此出发即可引出矛盾．实际上，任取一个复杂序列，从第二个符号开始，如果它与前一个符号不同，就将它改变成与它一样，否则就看下一个符号．这样的每一步动作都保持了所得到的序列为复杂序列的特点．由于序列长度有限，因此有限步后就得到完

全由同一个符号组成的简单序列. 这显然是个矛盾.

以上启发性的讨论告诉我们，关于符号序列的复杂性刻划不可能是非常简单的. 实际上，只是在60年代，利用了计算理论中的大量成果之后才形成了下面要介绍的定义.

§22.2 关于随机性的讨论

这里简要地说明一下，关于单个符号序列的复杂性或随机性的刻划问题与概率论中的随机性概念没有直接关系.

在概率论中生成随机序列的最容易的例子就是掷硬币试验. 将一枚有正反面区别的均匀硬币掷n次，在落地时将正面朝上记为1，反面朝上记为0，就可以得到一个长度为n的符号序列. 从经验就可以知道，在n充分大时，出现符号1的次数与出现符号0的次数一般是比较接近的. 但这里绝对没有排除连续多次出现正面朝上或相反的可能性. 从理论上这是独立随机变量序列，它是最简单的一种随机过程，有完整的数学理论. 但每次试验得到的符号序列只是这个随机过程的一个样本. 在概率论中并没有对一个样本是否随机下定义. 就上述试验来说，由n次掷硬币可得到2^n个不同的符号序列. 根据概率论，它们中的每一个出现的概率都是2^{-n}. 换句话说，在本节开始所举的四个符号序列，如果作为掷硬币试验得到的结果来看待，则出现的可能性完全一样.

这表明在研究一个给定序列的复杂性刻划问题时，我们不应当与该序列的生成过程发生联系. 在数学上为随机的信源所生成的符号序列，在直观意义上未必是随机的.

相反的情况大量存在. 在计算中广泛使用的伪随机数发生器即是一个例子. 它生成的序列完全是确定性的，但却可以通过各种关于随机性的检验要求. 另一个例子是取适当参数值的二次方映射或帐棚映射，后者可以直接用作随机数发生器. 动力系统的一个重要发现就是，一个完全是确定性的简单系统可以产生看起来完全为随机的输出(参看[22]等文献).

§ 22.3 描述复杂性

对于一个给定的符号序列，如何定义其复杂性或随机性，这在历史上并不是新问题，但在 60 年代之前始终没有令人满意的方法能解决这个问题.

在 60 年代中差不多同时，有三个人分别独立地提出了相同的方法. 他们是柯尔莫哥洛夫、恰依丁 (G.J.Chaitin) 和索洛莫诺夫(R. J. Solomonoff). 这种复杂性的定义在目前有许多不同的名称，与文献[105]相一致我们称它为柯尔莫哥洛夫复杂性. 这方面的原始文献是[106]～[108]. [105]是一个详细的综述，包括问题的历史和理论的多方面应用，并附有详细的文献.

先从一些简单情况说起. 设想要将一个符号串作为信息传送出去，考虑是否有比较经济的传送方法. 如果要传送的是长为 n 的全 0 串，则可以将"打印 n 次符号 0"这句话发送出去. 这里除了"n"这个信息之外，其他字都可以按某种固定方式编成符号串，其长度与 n 无关. 如果符号集为 $S=\{0, 1\}$，则大致用 $\log n$ 位二进制数就可以将 n 转化成 S^* 中的符号串. 例如，从 $2^{10}=1024$ 可知，用不到十个的 0、1 符号就可以将一千以下的 n 表示出来.

于是用 $\log n+c$ 位长度的 0、1 符号串就可以将长度为 n 的全 0 串这个信息传送出去. 其中 c 是与 n 无关的常数. 当 n 充分大时，这个方法与直接将这个符号串原封不动传送出去的方法相比，当然要好得多. 对于在本节一开始给出的第二个例子，即 10 串的多次重复，也可以采用类似的方法处理.

如果要传送的符号串看上去杂乱无章，找不出什么规律，则只能采取原原本本将它发送出去的方法.

由此可见，问题在于对一个给定的符号串有没有某种比较简单的描述方法. 如果可以用比原来的符号串短得多的方法来描述它，那么只要将这个描述本身(转化为符号串)发送出去就可以了. 反之，如果对于一个符号串的描述只能是其自身，则除了将它原原

本本发送出去之外就没有更好的方法了.

在这个意义上，我们下面要介绍的柯尔莫哥洛夫复杂性即是一种描述复杂性. 实际上,这也是它在文献中使用的名称之一.

§22.4 柯尔莫哥洛夫复杂性的定义

利用计算机可以将上面谈论的描述具体化为一个程序，将它输入到计算机中就能产生出给定的符号序列. 为简单起见，假定符号序列和程序都只用 0 和 1 两个符号表示. 程序本身也是符号串.

设给定的符号串为 x,将产生 x 的程序记为 p. 对一个计算机来说,p 是输入，x 是输出. 关于一个符号串 x 的柯尔莫哥洛夫复杂性就是产生 x 的最短程序 p 的长度. 它反映了对符号串 x 的最经济的描述所需要的符号个数. 可以记有关的计算机为 T，而将上述定义写为

$$K_T(x)=\min\{|p|\,|p\text{ 为产生 }x\text{ 的程序}\}.$$

如这样的 p 不存在,则令 $K_T(x)=\infty$. 但这个定义与计算机 T 有关. 容易理解，刻划一个符号串 x 的复杂性的方法不应当依赖于所用的计算机. 下面将要证明，存在一种具有某种“客观性”的计算机,可以克服上述定义的缺点.

这里需要在§2.4 中介绍的图灵机和通用图灵机的概念. 我们知道，可以将图灵机全体按照某种方式编号成符号序列. 记 $n(T)$是图灵机 T 的编号，它本身已成为一个由 0、1 组成的符号串. 将上面的定义中的计算机都看成为某个图灵机,按照丘奇-图灵论题这都没有问题.

现在设 U 是一个通用图灵机，它可以模拟任何一个图灵机的动作，并相应地定义 $K_U(x)$，它是在通用图灵机上生成串 x 并停机的最短输入的符号串长度.

我们考察 $K_U(x)$与 $K_T(x)$之间的差,其中 T 是任何一个图灵机.

设 p 就是在图灵机 T 上生成串 x 的最短程序，即有

$$|p|=K_T(x).$$

现在设 T 的编号为 $n(T)$. 我们可以将符号串 $0^{n(T)}1p$ 输入到上述通用图灵机 U 中，要求 U 在接受 $0^{n(T)}1$ 之后的动作完全模拟 T 的动作. 这样一来，由于 T 在输入 p 时能生成符号串 x 并停机，因此 U 在输入 $0^{n(T)}1p$ 时也能生成符号串 x 并停机. 根据定义，就有

$$K_U(x)\leqslant n(T)+1+|p|=n(T)+1+K_T(x).$$

记 $n(T)+1=c$，它是只与图灵机 T 有关的一个常数. 这样就得到

$$K_U(x)\leqslant K_T(x)+c.$$

这表明，用通用图灵机 U 得到的关于串 x 的复杂性 $K_U(x)$，与用其他图灵机 T 得到的复杂性 $K_T(x)$ 相比较，差别不超过只与 T 有关的一个常数. 在这个意义上，我们说 $K_U(x)$ 反映了符号串 x 本身的内在性质，即是对 x 的复杂性的一个客观度量.

还要考虑选择不同的通用图灵机所带来的差别. 容易与上面一样证明，如果 U' 是另一个通用图灵机，则存在一个只与 U 和 U' 有关的常数 $c_{UU'}$，使不等式

$$|K_U(x)-K_{U'}(x)|\leqslant c_{UU'}$$

对任何符号串 x 成立.

由此可见，我们可以在今后取定一个通用图灵机 U，并将 $K_U(x)$ 简记为 $K(x)$. 我们称这样的 $K(x)$ 是符号串 x 的柯尔莫哥洛夫复杂性.

§23 $K(x)$ 的性质与应用

在这一节中介绍柯尔莫哥洛夫复杂性 $K(x)$ 的一些性质和有关应用. 关于这方面的全面论述可参看[105].

§23.1 $K(x)$ 的基本性质

由于 U 是通用图灵机，我们容易看出 $K(x)$ 一定是有限数. 实

际上可以得到更好的结果. 设 T 是将输入串直接按原样输出并停机的一个特殊的图灵机. 记 $n(T)$ 为 T 的编号. 将 $0^{n(T)}1x$ 输入到通用图灵机 U 中, 就得到不等式

$$K(x) \leqslant |x| + c,$$

其中的常数 $c = n(T) + 1$, 与串 x 无关.

其次, 考虑长度 $|x| = n$ 的所有 2^n 个不同的 0、1 符号串. 由于长度小于 n 的程序 p 的个数不会超过

$$1 + 2 + 2^2 + \cdots + 2^{n-1} = 2^n - 1,$$

由此可见, 在 2^n 个长度为 $|x|$ 的串中, 至少有一个串的复杂性不小于 $|x|$, 即有

$$K(x) \geqslant |x|$$

成立. 在这个意义上, 我们说这个串 x 是不可压缩的, 也就是不存在更短的描述.

完全同样地可以证明, 在长为 n 的 2^n 个不同符号串中, 至少有二分之一的串的复杂性不低于 $n-1$, 至少有四分之三的串的复杂性不低于 $n-2$, …. 现设 n 充分大, 可以计算出在长度为 n 的串中复杂性低于 $n-10$ 的串的个数不超过

$$2^{n-10} - 1 = 1 + 2 + 2^2 + \cdots + 2^{n-11}.$$

将这个数与总数 2^n 相比, 可见在相同长度的符号串中, 有 99.9% 以上的符号串的最短描述只能压缩不到 10 个符号. 当 n 充分大时, 这个压缩量是微不足道的.

定义单个序列的随机性的一种方法是, 当 $|x| = n$, 且成立

$$K(x) \geqslant n - O(\log n)$$

时, 称串 x 为随机串.

我们在下面将会看到, 这个定义并非可以用于真正的计算. 但它已经表明, 在长度为 n 的符号串中, 当 n 充分大时, 99.9% 以上的串都是随机的. 这与我们的直观概念完全一致. 例如在 § 22.2 的掷硬币试验中, 只要 n 充分大, 所得到的符号串在直观上总是杂乱无章的随机串.

在[109]中发展了关于无限符号串的随机性定义，本书不作介绍.

更一般地，可以定义 g-不可压缩性，或者 g-随机性. 设 $g(n)$ 是一个整数值函数. 称长度为 n 的串 x 为 g-不可压缩，如满足不等式

$$K(x) \geqslant n-g(n).$$

与上面完全一样可以证明，在长度为 n 的符号串中间，复杂性低于 $n-g(n)$ 的串所占的比例不超过 $2^{-g(n)}$. 如果 g 具有性质

$$\lim_{n\to\infty} g(n)=\infty,$$

则这个比例趋于 0.

已经证明，当 n 充分大时，不可能证明任何一个符号串的柯尔莫哥洛夫复杂性大于 n. 在 [105]、[110]、[111] 中指出，这乃是哥德尔不完全性定理 (Gödel Incompleteness Theorem) 在信息论中的表现形式. 这里涉及到形式系统等方面的深刻内容，本书不再介绍. 这里只用于说明，要证明某些符号串是非随机是可能的. 例如，在上节一开始所说的两个序列当 n 充分大时可以有很经济的描述方法. 又如 π、e 等常数的近似值，当位数充分长时也是可以大大压缩的，因为可以用一个相对很短的程序来生成它们. 但另一方面，要证明关于一个给定的符号序列的任何描述都几乎与序列具有相同长度，则是完全不同的任务. 应用哥德尔定理得到的结果是，已经证明，当符号序列的长度 n 充分大时，这样的证明是不存在的. 注意这里在同一句话出现了两个"证明"，所指的含义不同.

§23.2 在自然数集上定义的 $K(x)$

设 x 为自然数，并将 x 与按字典排列的符号串序列

0, 1, 00, 01, 10, 11, 000, 001, ⋯

中的第 x 个等同起来. 然后将 $K(x)$ 中的自变量按上述方式理解，

这样就得到了在所有自然数上有定义的整数值函数 $K(x)$. 这一小节介绍函数 $K(x)$的一些基本性质.

用记号$|x|$表示自然数 x 所对应的上述符号串的长度(而不是 x 的绝对值). 于是可见有 $|x|\sim\log x$.

这时可建立函数 $K(x)$的下列性质:

1. $K(x)\leqslant|x|+c$,其中 c 为与 x 无关的常数.

2. $|x|=l$ 时满足不等式 $K(x)<l-m$ 的 x 所占的比例不超过 2^{-m}.

3. $\lim\limits_{x\to\infty}K(x)=\infty$. 由于长度有界的描述(即程序)只有有限个,因此它们所描述的 x 也只有有限多个.

4. 定义 $m(x)=\min\limits_{y\geqslant x}K(y)$,则 $m(x)$是单调增加的整数函数,$m(x)\leqslant K(x)$. 可以看出, $m(x)$ 是具有这两个性质的最大函数. 同样可以证明 $\lim\limits_{x\to\infty}m(x)=\infty$.

5. $K(x)$不是单调增加函数. 从 $K(x)$ 的定义已可看出, 存在 $m<n$, 但 $K(m)>K(n)$. 此外, 将 x 看成符号串时, $K(x)$ 对 x 自身的前缀也不具有单调性. 例如, 对上述一对自然数 m、n, 令 $x=0^n$, $y=0^m$, 则 y 是 x 的前缀, 但有 $K(0^n)<K(0^m)$. 这也就是说, 部分的复杂性可以超过整体的复杂性.

6. $K(x)$是不可计算函数. 这里需要在 § 2.5 中介绍的概念. 可以证明, $K(x)$不是部分递归函数, 因此在这个意义上是不可计算的. 不仅如此, 任何定义域为无限集的部分递归函数都不可能在自己的定义域上与 $K(x)$相等.

§ 23.3　$K(x)$在动力系统中的应用

勃鲁特诺(A. A. Brudno)将柯尔莫哥洛夫复杂性用于研究动力系统的轨道复杂性(见 [112]、[88]). 这里只限于对于单峰映射情况简述他的研究.

设 f: [0, 1]→[0, 1] 是§ 5 中介绍的单峰映射. 记从 x 出发

的轨为

$$f^*(x)=(x,\ f(x),\ \cdots,\ f^n(x),\ \cdots),$$

踪迹为

$$I(x)=A(f^*(x))=(s_0s_1\cdots s_n\cdots).$$

对于从 x 出发的轨，定义它的复杂性为

$$\overline{K}(x)=\overline{\lim_{n\to\infty}}\frac{1}{n}K(s_0s_1\cdots s_{n-1}),$$

其中 $K(s_0s_1\cdots s_{n-1})$是序列 $s_0s_1\cdots s_{n-1}$ 的柯尔莫哥洛夫复杂性.

在[112]中证明，如果 μ 是关于 f 的一个遍历测度，那么对于 μ-几乎处处 x，成立

$$\overline{K}(x)=h_\mu,$$

这里的 h_μ 是关于 μ 的测度熵. 实际上，在 [112] 中对于相当一般的动力系统证明了这个结论.

如果 f 满足施瓦兹导数为负的条件，则在[88]中证明，对于在勒贝格测度意义下的几乎所有 x，$\overline{K}(x)$是一个常数. 将这个常数记为 $\overline{K}$，又可以证明，在李雅普诺夫指数 $\lambda>0$ 时，成立

$$\overline{K}=\lambda.$$

对于 f 有稳定周期轨或为费根鲍姆吸引子的情况，则已证明有

$$\overline{K}=0.$$

由此可见，将柯尔莫哥洛夫复杂性应用于动力系统所得到的结果，与李雅普诺夫指数和测度熵在很多情况是一致的.

在[88]中指出，$\overline{K}$ 在某些情况是关于动力系统复杂性的一个更为深入的刻划. 已经证明，对于满足不等式

$$0<h<\frac{1+\sqrt{5}}{2}$$

的每一个 h，存在满足施瓦兹导数为负条件的单峰映射 f，它的李雅普诺夫指数 $\lambda=0$，但却有 $\overline{K}=h$ 成立. 这样的 f 在二次方映射族中也是存在的.

§23.4 在形式语言中的一个应用

介绍柯尔莫哥洛夫复杂性在形式语言中的一个应用，它在很多情况中可以代替在§1.7中介绍的泵引理.

定理 如$\mathrm{L}\subseteq S^*$为正规语言,则有一个只与L有关的常数c,使得对于每一个符号串$x\in S^*$,如果

$$y\in \mathrm{L}_x=\{y \mid xy\in \mathrm{L}\},$$

而且y在L_x中按字典方式排序时是第n个串,那么成立

$$K(y)\leqslant K(n)+c.$$

我们给出这个结果的证明,然后举例说明它的应用.

证明的关键是考虑y的描述. 由于L是正规语言,设M是接受L的确定性有限自动机. 设在输入x之后自动机处于状态q. 从q出发，按字典方式顺序检查使自动机进入终止状态的第n个符号串当然就是y. 这就提供了对y的一个描述，可以用一个图灵机来实现这个描述,它能模拟上述有限自动机,同时能按字典方式的顺序搜索y. 在这个描述中，信息n是必需的. 它的描述为$K(n)$. 最后,利用q只有有限个，就可以求出与x无关的常数c,得到所要的不等式.

可以看出，从本质上来说这个结果与本书在§1.8介绍的迈希尔-奈罗德定理是一致的.

现在举一个例子. 这时用泵引理是不方便的. 设$S=\{0\}$,语言

$$\mathrm{L}=\{0^p \mid p\text{ 为素数}\}.$$

我们证明L不是正规语言.

设p'是第k个素数,p是第$k+1$个素数. 我们知道$p-p'$的最小值为2，即p'和p为孪生素数的情况. 但容易证明$p-p'$没有上界. 例如,$n!+2, n!+3, \cdots, n!+n$都不是素数，其中n是任意自然数. 由它们隔开的两个素数之差就不小于n.

令$x=0^{p'}$,考虑集合

$$L_x = \{y \mid xy \in L\}.$$

在其中 $y=0^{p-p'}$ 是第一个串．利用上面的结果，以及目前有 $n=1$，就得到

$$K(p-p') = O(1),$$

右方是一个由语言 L 确定的常数．由于长度有界的描述只有有限多个，这与 $p-p'$ 无界相矛盾．

柯尔莫哥洛夫复杂性已有多方面的应用，可参看[105]．

§24 基于移位寄存器的复杂性

本节介绍以移位寄存器为基础的复杂性刻划方法．材料主要来自[113]～[120]．

§24.1 移位寄存器序列

考虑符号序列

$$s = s_0 s_1 \cdots s_i \cdots,$$

它可以是有限或无限序列．如果存在自然数 n，使得在序列 s 中从第 $n+1$ 个符号开始，每一个符号都可以由在它之前的 n 个符号确定，即满足递归关系

$$s_i = f(s_{i-n}, s_{i-n+1}, \cdots, s_{i-1}), \quad i \geq n,$$

则称 s 为(反馈)移位寄存器序列．它可以用图 24-1 中的(反馈)移位寄存器来实现，其中有 n 个寄存器，我们按从右到左的顺序称它们为第 1 级、第 2 级、…、第 n 级寄存器．每一级寄存器都可以处于有限多个状态中的任何状态．这对应于符号串 s 的每一个符号均取自一个有限符号集 S．称 n 为移位寄存器的长度．



图 24-1

在寄存器开始工作时，这 n 级的初态从左到右为

$$s_{n-1},\ s_{n-2},\ \cdots,\ s_1,\ s_0.$$

从处于最右边的第 1 级寄存器读出的内容即作为这 n 级移位寄存器的输出.

同时这 n 级寄存器的状态经过图 24-1 中上方的一个矩形所代表的开关线路进行综合. 在一个移位脉冲的作用下，每一级寄存器中的内容传递给右边的一级寄存器，同时将上述开关线路的输出传递给第 n 级寄存器，即图 24-1 所示最左边的寄存器.

在第一个移位脉冲作用下，n 级移位寄存器的状态从左到右变为

$$s_n,\ s_{n-1},\ \cdots,\ s_2,\ s_1.$$

第 1 级寄存器的原先状态 s_0 已被读出，作为输出的第一个符号.

按照上述方式不断加以移位脉冲，就可以输出序列 s. 我们也说这个序列 s 由上述移位寄存器生成.

在图 24-1 中的开关线路所实现的输入输出关系即本节一开始写出的递归关系 f. 我们今后也称 f 为开关函数，它是从 S^n 到 S 的映射. 以下一般均取 $S=\{0,\ 1\}$来讨论. 如果 f 为线性函数，即可写为

$$s_i=-c_1s_{i-1}-c_2s_{i-2}-\cdots-c_ns_{i-n},\qquad i\geqslant n,$$

则称序列 s 为线性(反馈)移位寄存器序列. 当然，这里的乘法与加法都在与 S 相应的有限域中进行. 对于 $S=\{0,\ 1\}$，上述等式右方的系数 c_i 为 0 或 1，运算按模 2 进行. 相应地可以用图 24-2 的线性(反馈)移位寄存器来生成序列 s. 容易理解，线性移位寄存器在技术上是非常简单的. 但是应当看到，不同的 n 级线性移位寄存器有 2^n 个，但不同的 n 级移位寄存器则有 2^{2^n} 个，因此提供了多得多的可能性.

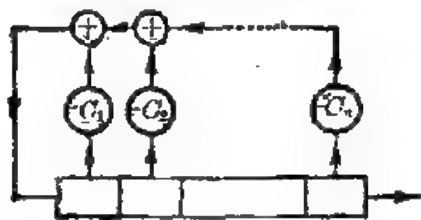


图 24-2

现在叙述以移位寄存器为计算模型的序列复杂性的度量方法.

对于给定的有限序列

$$s = s_0 s_1 \cdots s_{N-1},$$

将生成 s 的长度最短的线性移位寄存器的级数称为序列 s 的线性复杂性(或线性复杂度)。相仿地在允许 f 为二次或 k 次函数时，可以定义 s 的二次复杂性或 k 次复杂性. 在 [117] 中将生成 s 的级数最少的移位寄存器的级数称为序列 s 的极大复杂性 (Maximal Order Complexity).

以下主要介绍比较成熟的线性复杂性. 注意这时在多数文献中所称的序列 s 的线性复杂性是指生成周期序列 s^∞ 的最短线性移位寄存器的级数.

§24.2 几个简单例子

[例 1] 如图 24-3 所示是一个四级线性移位寄存器. 相应的递归关系(或开关函数)为

$$s_i = s_{i-1} + s_{i-2} + s_{i-3} + s_{i-4}, \qquad i \geq 4.$$

可以证明，根据不同的初态，它有四种输出，即全 0 串和三组周期 5 的周期序列，即

$$(00110)^\infty, \quad (11110)^\infty, \quad (01010)^\infty,$$

以及它们的所有移位序列.

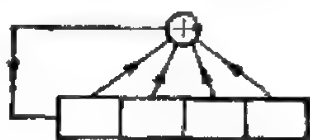


图 24-3

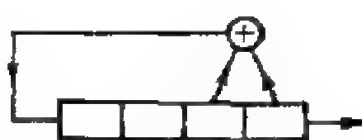


图 24-4

[例 2] 图 24-4 是另一个四级线性移位寄存器，递归关系为

$$s_i = s_{i-3} + s_{i-4}, \qquad i \geq 4.$$

可以证明，除了生成全 0 串之外，这个移位寄存器的输出是周期 15 的周期序列，即 $(000100110101111)^\infty$ 及其所有移位序列.

在图 24-2 所示的 n 级线性移位寄存器的一般框图中，如系数 $c_n \neq 0$，则称为是非退化的. 上面的两个例子都是非退化的.

可以证明，非退化的线性移位寄存器在任何初始状态下的输出总是周期序列．容易由此看出，如不加上非退化限制，则输出总是终极周期序列．这对于一般的移位寄存器也成立，可参见[113]．

§24.3 线性复杂性的计算方法

以下对 $S=\{0, 1\}$ 介绍线性复杂性的计算方法．全面的叙述与证明参见[113]、[114]．

对于满足递归关系

$$s_i=-c_1s_{i-1}-\cdots-c_ls_{i-l}, \qquad i\geqslant l$$

的线性移位寄存器序列，引入联系多项式

$$f(x)=1+c_1x+\cdots+c_lx^l,$$

称 l 为寄存器的级数，并采用记号 $\langle f, l\rangle$．

现设给定长度 N 的一个有限序列

$$s=s_0\ s_1\ \cdots\ s_{N-1}.$$

计算的方法是对每个 $n=1, 2, \cdots, N$，求出生成 s 的前缀 $s_0\ s_1\ \cdots\ s_{n-1}$ 的最短线性移位寄存器．记相应的联系多项式为 f_n，级数为 l_n．采用递归方法，从 $n=1$ 开始计算到 $n=N$．

现分别叙述算法的开始部分和递归部分：

1．如 $n_0\geqslant 0$ 使

$$s_0=s_1=\cdots=s_{n_0-1}=0, \qquad s_{n_0}=1,$$

则令

$$d_0=d_1=\cdots=d_{n_0-1}=0, \qquad d_{n_0}=s_{n_0},$$
$$f_1=f_2=\cdots=f_{n_0}=1, \qquad f_{n_0+1}=1+x^{n_0+1},$$
$$l_1=l_2=\cdots=l_{n_0}=0, \qquad l_{n_0+1}=n_0+1.$$

2．设已求出 $\langle f_i, l_i\rangle$，$i=1, \cdots, n$，$n_0<n<N$，其中

$$l_1=\cdots=l_{n_0}<l_{n_0+1}\leqslant l_{n_0+2}\leqslant\cdots\leqslant l_n.$$

写出

$$f_n=1+c_{n1}x+c_{n2}x^2+\cdots+c_{nl_n}x^{l_n}.$$

计算

$$d_n=s_n+c_{n1}s_{n-1}+\cdots+c_{nl_n}s_{n-l_n}.$$

如果 $d_n=0$，则令

$$f_{n+1}=f_n, \qquad l_{n+1}=l_n.$$

即生成 $s_0s_1\cdots s_{n-1}$ 的 $\langle f_n, l_n\rangle$ 也能生成 s_n. 如果 $d_n\neq 0$, 设 m 满足条件 $1\leqslant m<n$, $l_m<l_{m+1}=\cdots=l_n$, 令

$$f_{n+1}=f_n+x^{n-m}f_m, \qquad l_{n+1}=\max\{l_n, n+1-l_n\}.$$

采用以上公式计算, 直到 $\langle f_N, l_N\rangle$ 为止, 这时的 f_N 即是最短线性移位寄存器的联系多项式, l_N 是其级数, 也就是序列 s 的线性复杂性.

如果要求生成周期为 N 的无限序列 $(s_0s_1\cdots s_{N-1})^\infty$ 的最短线性移位寄存器, 则可以证明, 只要用同一算法于序列 $(s_0s_1\cdots s_{N-1})^2$, 求出 $\langle f_{2N}, l_{2N}\rangle$, 就可以得到所要的结果.

[例 1] 计算生成长度为 15 的序列

000100110101111

的线性复杂性. 将算法中确定 $\langle f_n, l_n\rangle$ 的计算称为第 n 步计算. 整个过程如下:

前四步为开始部分. 从

$$s_0=s_1=s_2=0, \qquad s_3=1$$

得到

$$d_0=d_1=d_2=0, \qquad d_3=1,$$
$$f_1=f_2=f_3=1, \qquad f_4=1+x^4,$$
$$l_1=l_2=l_3=0, \qquad l_4=4.$$

第五步: $d_4=s_4+s_0=0$, $f_5=f_4$, $l_5=l_4$.

第六步: $d_5=s_5+s_1=0$, $f_6=f_5$, $l_6=l_5$.

第七步: $d_6=s_6+s_2=1$.

这时 $n=6$, $m=3$, 得到

$$f_7=f_6+x^3f_3=1+x^3+x^4, \qquad l_7=4.$$

在下面的计算中 $d_n(n\geqslant 7)$ 始终为 0, 因此 $\langle f_7, l_7\rangle$ 就是生成上述序列的最短移位寄存器. 实际上这即是图 24-4 中的结果.

§24.4 特布里渊序列

本小节介绍由特布里渊(N. G. deBruijn)提出而命名的特布里渊序列, 它有广泛的应用(参见[118]、[115]、[116]).

特布里渊序列是由 0 和 1 组成的周期为 2 的幂 2^n 的周期序列，在它的长为 2^n+n-1 的子串中出现 2^n 个所有长 n 的 0、1 组合，即每一组合恰好出现一次. $n=1$、2、3 时的特布里渊序列的一个周期为：

$$n=1,\qquad 01,$$

$$n=2,\qquad 0011,$$

$$n=3,\qquad 00111010 \text{ 与 } 01110001.$$

可以用一个有向图(或状态转移图)来研究特布里渊序列. 将 2^n 个长为 n 的不同串用作图中的 2^n 个结点. 从结点$(x_1x_2\cdots x_n)$到结点$(y_1y_2\cdots y_n)$存在有向弧的条件是

$$y_1\cdots y_{n-1}=x_2\cdots x_n,$$

这时就用一条标以符号 y_n 的有向弧连接这两个结点. 这样就使每个结点有两条弧通向其他结点. 特布里渊序列对应于在这个有向图中的一条封闭回路，它从任何一个结点出发访问每一个结点恰好一次后回到起点.

一般称上述有向图为特布里渊图. 在本书的某些部分已使用这类似的图(见图 15-1和图 19-1).

可以证明，上述不同闭路的个数为 $2^{2^{n-1}-n}$ 个(参见 [118]). 在表 24.1 上列出了对应于前几个 n 的特布里渊序列的周期、个数以及下面将要介绍的线性复杂性 $c(s)$ 的上、下界. 这里对于移位后相同的各个周期序列只计为一个.

表 24.1

n	1	2	3	4	5	6
周　期	2	4	8	16	32	64
个　数	1	1	2	16	2048	67108864
$c(s)$	2	3	7	12～15	21～31	38～63

已经证明(见[115]、[116])，在 $n\geq 3$ 时，周期为 2^n 的特布里渊序列的线性复杂性 $c(s)$ 满足不等式

$$2^{n-1}+n\leqslant c(s)\leqslant 2^n-1,$$

同时这里的上、下界均可达到.

在 $n=3$ 时，两个周期 8 的特布里渊序列的 $c(s)$ 的上述上、下界恰好相等，都等于7. 利用在§24.2中的算法即可求出在图24-5中所示的最短线性移位寄存器. 只要取合适的初态就可以分别生成这两个特布里渊周期序列. 相应的联系多项式是



图 24-5

$$1+x+x^2+x^3+x^4+x^5+x^6+x^7.$$

一般认为，特布里渊序列具有较高的线性复杂性，可以作为复杂序列的很好的有限逼近. 近来也有关于它的 2 次复杂性与 $k(>2)$ 次复杂性的研究. 但从上述特布里渊图可以知道，周期 2^n 的特布里渊序列只要用 n 级非线性移位寄存器就可以生成. 例如，生成 $n=3$ 的特布里渊序列

$$00111010$$

的开关函数为：

$$111\to 0,\quad 110\to 1,\quad 101\to 0,\quad 010\to 0,$$
$$100\to 0,\quad 000\to 1,\quad 001\to 1,\quad 011\to 1.$$

因此按照在 [117] 中的极大复杂性度量，特布里渊序列为简单序列，或者说复杂程度不高的序列.

§24.5 与 $K(x)$ 的比较

回顾关于柯尔莫哥洛夫复杂性的定义，可以认为在本节介绍的复杂性也是一种描述复杂性，只是在那里是以通用图灵机为计算模型，而在本节则用移位寄存器为计算模型.

实际上，设一个有限序列 s(这里即指周期序列 s^∞)的线性复杂性为 $c(s)$，那么从最短线性移位寄存器的 $c(s)$ 个系数和 $c(s)$ 个初始状态 $s_0s_1\cdots s_{c-1}$，就给出了 s 的一个描述. 从这个观点出发，在

[119]中提出将 $\Lambda(s)=2c(s)$ 作为关于符号串 s 的新的复杂性度量，并与柯尔莫哥洛夫复杂性 $K(s)$ 作比较.

在[119]中证明：

1. 对任何给定的 ε, $0<\varepsilon<1$,

$$\text{prob}\{s\in\{0,1\}^n \mid (1-\varepsilon)\Lambda(s) \leqslant K(s)\leqslant(1+\varepsilon)\Lambda(s)\}\to 0(n\to\infty).$$

2. 如 s 为无限序列，$s^{(n)}$ 是 s 的长为 n 的前缀，那么在勒贝格测度的意义上几乎处处成立

$$\lim_{n\to\infty}\frac{K(s^{(n)})}{\Lambda(s^{(n)})}=1.$$

由这两个结果可以看出，对于绝大多数符号串来说，这两种复杂性度量方法似乎是一致的. 在[120]中对此提出了不同的观点，认为仅仅从以上两点不能得出上述结论，并且指出了线性复杂性与柯尔莫哥洛夫复杂性很不相同的一些特点. 这些问题有待今后作进一步研究.

§25 兰帕尔–齐夫复杂性

本节介绍兰帕尔和齐夫(A. Lempel, J. Ziv)提出的一种复杂性度量. 主要材料来源为他们的论文[121]及其在动力系统中的应用.

§25.1 一种容易计算的复杂性

我们已经知道，在§22定义的柯尔莫哥洛夫复杂性是不可计算的. 实际上，在上节介绍的线性复杂性在很多情况要作准确计算也是相当困难的. 例如，在§24.4中的特布里渊序列的线性复杂性分析就是在不久之前才得到完全解决的(参见[115]、[116]).

兰帕尔和齐夫在[121]中提出一种相对来说容易计算得多的

复杂性度量．在[91]的最后一章对此也作了介绍，并称之为兰帕尔-齐夫复杂性．此外，在某些文献中也将[121]中的复杂性称之为柯尔莫哥洛夫复杂性，实际上两者有很大不同，请读者注意．

为叙述方便起见，本节在叙述时如不作说明，一般均考虑由 $S=\{0, 1\}$ 所生成的有限或无限符号序列，将串 s 的复杂性记为 $c(s)$ 或 c．

仍然采取在§22中的描述复杂性的观点．但在这里代替图灵机而采用只具有两种简单操作的计算模型来描述一个给定序列，并将所需的某种操作次数作为序列的复杂性度量．

第一种操作是复制，或者说拷贝，简单来说，即用序列的某个前缀中的子串生成更长的前缀．第二种操作即是对已生成的前缀添加一个符号．当然，任何一个长度为 n 的符号序列可以从空串开始，用 n 次第二种操作生成．但我们所关心的是与§22一样的最短描述．在[121]中提出一个算法，它直接可以对给定序列作复杂性分析，给出最短描述．

先举几个例子，然后说明一般法则．

［例1］ 序列为全0串0000…．

从空串 ε 出发用添加操作生成第一个符号0．在这以后只需用复制方法即可．将这个过程记为

$$0000\cdots \to 0\cdot 000\cdots,$$

这里右方出现的"·"代表第二种操作的使用．按照[121]，重复使用复制操作时不用"·"分开．最后，将由记号"·"分成的子串个数定义为该符号串的复杂性度量．对于上述例子，只要0串的长度大于1，复杂性 c 都是2．

［例2］ 101010…．

先从 ε 开始添加1，然后再添加符号0．在这之后只要用复制方法就够了．这样就得到

$$101010\cdots \to 1\cdot 0\cdot 1010\cdots,$$

复杂性 $c=3$．

现在叙述一般法则. 设序列为

$$s_1s_2\cdots s_n.$$

从 e 出发开始添加 s_1. 现考虑中间步骤, 设已生成前缀 $s_1s_2\cdots s_{r-1}$, $r<n$, 并且下一个符号 s_r 是用添加操作完成的, 记为

$$s_1s_2\cdots s_n \to s_1\cdot s_2\cdots s_{r-1}s_r\cdot s_{r+1}\cdots.$$

这里在 s_r 后的记号"·"反映了 s_r 的生成过程. 现在给出如何做下去的法则.

先令 $Q=s_{r+1}$, 观察 Q 是否可以从 $sQ\pi$ 用拷贝方法得到, 其中 $s=s_1\cdots s_r$, π 表示将它前面的符号串的最后一个符号去掉后的结果, 在这里即是有 $sQ\pi=s=s_1\cdots s_r$.

如果 Q 不能从 $sQ\pi$ 中某个子串拷贝得到, 则就用添加操作加上 s_{r+1}, 并加上一个记号"·", 这样又回到与刚才相同的情况.

如果 $Q=s_{r+1}$ 可以从 $sQ\pi(=s)$ 中某个符号复制得到, 则继续观察 $Q=s_{r+1}s_{r+2}$ 能否从 $sQ\pi$ 的某个子串复制得到. 这时

$$sQ\pi=s_1\cdots s_rs_{r+1},$$

如果能办到, 则再考虑 $Q=s_{r+1}s_{r+2}s_{r+3}$, 并提出同样的问题. 这样下去有两种可能, 或者 Q 已包含了原来给定的序列的最后一个符号 s_n, 则分析已经结束. 或者对某个 Q, 它不再能从 $sQ\pi$ 的任何一个子串复制得到. 这时就采取添加操作, 将这个 Q 的最后一个符号添上, 并在它后面加上记号"·".

因此可以看到, 实际上记号"·"的个数反映了采取添加操作的次数. 如果符号串在上述分析结束时以"·"结束, 则这个记号"·"的个数就等于符号串的复杂性. 否则, 将个数加一即得到复杂性.

[例 3] 0010.

按照上述算法共有四步:

1. 从空串 e 加上 0, 记为 $\to 0\cdot$.

2. 这时 $s=0$, $Q=0$, $sQ=00$, $sQ\pi=0$. 因此 Q 可以从 $sQ\pi$ 复制得到, 记为 $\to 0\cdot 0$.

3. $s=0$, $Q=01$, $sQ=001$, $sQ\pi=00$. 显然 Q 不能从 $sQ\pi$ 复

制得到. 采用添加操作得到 $\to 0\cdot01\cdot$.

4. $s=001$, $Q=0$, $sQ=0010$, $sQ\pi=001$, Q 可从 001 拷贝得到. 计算已结束. 最后的结果是

$$0010\to 0\cdot01\cdot0.$$

复杂性 $c=3$.

[例 4] 0001101001000101.

这是在[121]中的例子. 我们只列出复杂性分析的最后结果为

$$0\cdot001\cdot10\cdot100\cdot1000\cdot101,$$

它的复杂性为 6.

以上分析方法容易在计算机上编成程序. 在[122]中有相应的方框图.

§25.2 理论基础

如果将符号串 s 的所有子串称为 s 所具有的词, 记其全体为 $v(s)$, 那么上述分析过程可以看成是不断寻找新词的过程. 具体来说, 如果某个中间过程为 $s\to sQ=R$, 那么当 $Q\in v(R\pi)$ 时, 就称 R 可以从 s 出发用复制操作生成. 否则就用添加操作. 这意味着在 R 中一定出现了在 $R\pi$ 中不具有的某些新词, 它(或它们)一定是 R 串的后缀.

理论上已经证明, 只要按照上一小节介绍的算法, 就可以得关于一个符号串的最短描述, 即添加操作次数尽可能少的描述.

现设符号集 S 有 α 个符号, 序列 $s\in S^*$ 的长度 $|s|=n$. 在[121]中证明, $c(s)$ 的上界为

$$c(s)<\frac{n}{(1-\varepsilon_n)\log n},$$

其中

$$\varepsilon_n=2\cdot\frac{1+\log\log(\alpha n)}{\log n},$$

它在 n 充分大时是一个小量, 且有 $\varepsilon_n\to0(n\to\infty)$. 因此我们一

般说 $c(s)$ 以 $n/\log n$ 为上界，其中 $n=|s|$，对数 log 以 α 为底.

与柯尔莫哥洛夫复杂性相同的是，当 n 充分大时，在长为 n 的所有 α^n 个不同符号串中，绝大多数串的复杂性都接近于这个上界 $n/\log n$.

如果假定这 α^n 个符号串都以相同的概率出现，则在 [121] 中证明，对每个 $\varepsilon\in(0, 1)$，成立

$$\lim_{n\to\infty}\operatorname{prob}\left\{c(s)<\frac{n(1-\varepsilon)}{\log n}\,\middle|\,|s|=n\right\}=0.$$

与 § 23.1 相仿，如果将 n 充分大时成立

$$c(s)\sim\frac{n}{\log n}$$

作为随机串的定义，那么可以看出，在 n 充分大时，长度为 n 的符号串中绝大多数都是随机串，或者说是复杂的. 反之则说在符号串 s 中有模式或结构存在.

我们现在将这个复杂性定义用于分析在 § 24.4 中介绍过的特布里渊序列. 这时 $S=\{0, 1\}, \alpha=2$.

[例 5] 设 s 是周期为 2^k 的特布里渊序列中长度为 2^k+k-1 的一个有限子串. 这时，如 § 24.4 所述，在 s 中的任何两个长为 k 的子串都是彼此不相同的，因此在用本节的复杂性分析时，从前一个记号"·"到下一个记号"·"之间的长度不可能超过 k. 于是对 $|s|=n$ 就有

$$c(s)\geqslant\frac{2^k+k-1}{k}=\frac{n}{\log(n-k+1)}\geqslant\frac{n}{\log n}$$

成立. 因此按照这里关于复杂性(即随机性)的定义，特布里渊序列具有较高的复杂性.

§ 25.3 关于非等概率情况的修正

在上述理论分析时的一个重要前提是，在 S 中的各个不同符号在符号串中是以等概率方式出现的. 对于给定的一个充分长的符号串或从某个信源发出的符号串，可以统计各个符号出现的实

际频率．如果发现明显不符合上述等概率前提时，在[121]中提供了下列修正．

限于讨论 $\alpha=2$ 的情况，即 $S=\{0,\ 1\}$，并用 p 和 $1-p$ 分别代表符号 0 和 1 的出现概率．这时信息论中给出了信源熵

$$h=-p\log p-(1-p)\log(1-p) \qquad (c<1)$$

的计算公式．在 $p=0.5$ 时 h 达到极大值 1．

在[121]中证明，只要将上一小节中的 $n/\log n$ 用 $hn/\log n$ 代替，有关结论仍然成立．

具体来说，先用统计方法求出各个不同符号出现的概率(的近似值)．计算熵 h．在 h 接近 1 时，用上一小节的结论．如果 h 明显小于 1，则将复杂性分析所得的度量 $c(s)$ 与 $hn/\log n$ 作比较．如两者很接近，我们认为符号串是复杂性较高的串，或者说是随机串．如果 $c(s)$ 明显低于 $hn/\log n$，则认为在串 s 中存在某种模式．

§25.4 在动力系统中的应用

在[122]中将兰帕尔-齐夫复杂性用于单峰映射与一维元胞自动机，得到了很有意义的结果．这里主要介绍在单峰映射中的应用情况．

与§23.3类似，先将轨道转化成符号序列，然后计算它(们)的复杂性，不同的轨可以有不同的复杂性．

设有踪迹 $I(x)=A(f^*(x))=s_1s_2\cdots s_n\cdots$（见§5.2），将它的长为 n 的前缀记为

$$s^{(n)}=s_1s_2\cdots s_n,$$

并且记

$$c_n=c(s^{(n)}).$$

然后，同时计算 c_n 和归一化的复杂性度量

$$I_n=\frac{c_n}{b_n},$$

其中 $b_n=n/\log n$．当然，在这之前应进行统计处理，以判断是否要

用 § 25.3 中的修正.

对于二次方映射 $f(x)=bx(1-x)$, $3\leqslant b\leqslant 4$, 在 [122] 中计算了 c_n 和 b_n 的值, 得到如图 25-1 所示的结果, 其中包含以下内容:

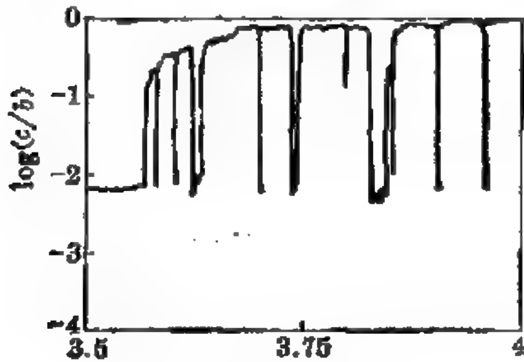


图 25-1

1° 对于有稳定周期轨的情况, 当 n 充分大时, c_n 趋于某个有限值.

2° 在费根鲍姆点 $b \doteq 3.57$ 处, 当 $n\to\infty$ 时 c_n 趋于无穷大, 但 $I_n=c_n/b_n$ 则趋于 0.

3° 在混沌情况, 当 $n\to\infty$ 时 $I_n=c_n/b_n$ 趋于有限值.

不难对于这些实验结果作出理论上的解释. 首先要注意, 虽然不同轨可具有不同的复杂性, 但对于二次方映射的上述几种情况, 除去一个勒贝格测度为零的集之外, 从其余的初始点出发的轨具有类似的符号动力学行为. 上述计算机实验是对这类典型轨来进行的.

对于有稳定周期轨的情况, 不妨直接讨论周期踪迹 s^∞. 与 § 25.1 中例 1、2 完全一样, 只要 $n\geqslant|s|$, 就有 $c_n=c(s)$. 对于一般的典型轨也是类似的. 因此当 n 充分大时 c_n 收敛于某个有限值.

对于费根鲍姆点的结果也是容易理解的. 为简单起见下面只分析揉序列本身. 对其他典型轨的分析可利用 § 10.4 中的结果来做. 这时我们知道, 有限时间内落到不稳定周期轨上的情况是非典型的, 而与吸引子距离趋于 0 的轨是典型的.

如 § 10.2 那样, 记 t_∞ 为这时的揉序列. 对于它的长为 n 的前缀, 存在 k, 使

$$2^{k-1}<n\leqslant 2^k$$

成立, 直接利用在§ 10 中的结果, 从

$$t_\infty=\lim_{n\to\infty} t_n, \qquad t_{n+1}=t_n\hat{t}_n \qquad (n\geqslant 0)$$

就得到
$$c_n=k+1.$$

这样就已得到

$$\lim_{n\to\infty} c_n = \infty.$$

另一方面，对于上述归一化复杂性可以建立

$$I_n = \frac{k+1}{\left(\frac{n}{\log n}\right)} \leqslant \frac{(k+1)\log 2^k}{2^{k-1}} = \frac{k(k+1)}{2^{k-1}},$$

因此得到

$$\lim_{n\to\infty} I_n = 0.$$

从这里看到，采用 c_n 和 I_n 两个度量来刻划费根鲍姆吸引子是有好处的。前者反映了它比有稳定周期轨情况更复杂的一个侧面，后者则反映出它的动力学行为也有简单的另一个侧面。实际上，这与 § 13.2 的分析是一致的。

对于粗粒混沌情况(参见[26])，利用在 § 9.5 中的结果，可以证明当 $n\to\infty$ 时 I_n 趋于有限正值。这里最简单的情况即是二次方映射中的满射情况，映射 $f(x)=4x(1-x)$ 存在绝对连续遍历不变测度，它的支集即是区间[0, 1]，典型轨于其中稠，因此它的踪迹是随机性达到极大的符号序列，于图 25-1 中表现为 $\lim_{n\to\infty} I_n=1$。对于一般的粗粒混沌也可以作出类似分析。

在[122]中还将这种复杂性分析方法研究一维元胞自动机，得到了很有意义的结果。这时由于兰帕尔-齐夫复杂性的计算量很小，可以计算多次迭代后的构形所具有的复杂性，这是一个很大的优点。

关于这种复杂性度量在动力系统研究中的进一步应用与分析，还有待今后的探索。

附录 A

本书§6中两个定理的证明

在这个附录中给出在§6.1的定理1和§6.5的定理3的数学证明.

A.1 定理1的证明

为读者方便起见,将定理1叙述如下,

定理1 设f为单峰映射, KS为f的揉序列,语言

$$\mathrm{L}=\mathscr{L}(\mathrm{KS}),$$

那么对任何$t\in \mathrm{L}$,存在点$x\in[0, f(c)]$,使t是踪迹$I(x)$的前缀.

证明 分KS不含符号c和含符号c的两种情况来讨论. 在证明中经常要将有限序列与另一个有限或无限序列比较, 分出序的大小. 这里的定义与§5.3相同. 但在等号成立时, 则应理解为较短序列为较长序列(包括无限序列)的前缀.

用反证法. 设存在一个序列$t\in \mathrm{L}$, 使得对任何点$x\in[0, f(c)]$, t都不是踪迹$I(x)$的前缀.

先讨论揉序列KS不含符号c的情况.

定义集合

$$L_t=\{x\in[0, f(c)]\,|\,I(x)<t\}$$

和
$$R_t=\{x\in[0, f(c)]\,|\,I(x)>t\}.$$

我们将证明L_t和R_t都是区间$[0, f(c)]$中的非空开集. 由反证法假设,就有$[0, f(c)]=L_t\cup R_t$, 引出矛盾. 以下只讨论L_t. 对R_t的讨论是类似的.

由反证法假设,可见t不会是全0串,因此$0\in L_t$. 同样可知$f(c)\in R_t$. 因此两个集合都是非空集.

设 $t=t_1t_2\cdots t_n$，其中 n 是串 t 的长度. 由语言 L 的定义，在 $t(\in \mathrm{L})$ 中不含符号 c.

任取 $y\in L_t$. 记 $I(y)=s_1s_2\cdots s_n\cdots$. 我们要证明存在一个邻域 $O(y)\subseteq L_t$，即证明 y 为 L_t 的内点. 这样就可证明 L_t 为非空开集.

从 $y\in L_t$，有 $I(y)<t$. 因此存在 l，$1\leqslant l\leqslant n$，使 $s_1\cdots s_{l-1}=t_1\cdots t_{l-1}$，$s_l\neq t_l$. 我们知道 t_l 不是 0 就是 1. 以下讨论 $t_l=1$ 的情况. 对于 $t_l=0$ 的讨论是类似的. 下面将分别几种情况来构造 y 的邻域 $O(y)$.

从 $t_l=1$，可见 $s\cdots s_{l-1}$ 是偶串，s_l 为 0 或 c. 如果 $s_l=0$，则利用映射 f 连续，可以定义

$$O(y)=\{x\in[0,\ f(c)]\mid I(x)\text{ 以 }s_1\cdots s_l\text{ 为前缀}\}.$$

这时的 $O(y)$ 为开区间，且满足 $O(y)\subseteq L_t$.

对于 $t_l=1$ 和 $s_l=c$ 的情况，可以看出 $l<n$（而不能有 $l=n$）. 否则，考虑开区间

$$\{x\in[0,\ f(c)]\mid I(x)\text{ 以 }s_1\cdots s_{l-1}\text{ 为前缀}\}.$$

在这个区间上 f^{l-1} 单调. 从 $s_l=c$，可见 f^{l-1} 取到临界点 c，从而也会取到在临界点两侧的值. 因此，如有 $l=n$，则已找到一个点 x，使 $I(x)=t$，与反证法假设矛盾.

从 $l<n$ 出发，可以考虑 t 的后缀，它从 t_{l+1} 开始. 同时，由于从 s_{l+1} 开始的符号串一定是揉序列本身，因此成立

$$t_{l+1}\cdots t_n\leqslant s_{l+1}\cdots s_n.$$

如果成立严格不等号，则存在 α，使有 $l+\alpha\leqslant n$ 和

$$t_{l+1}\cdots t_{l+\alpha-1}=s_{l+1}\cdots s_{l+\alpha-1},\qquad t_{l+\alpha}\neq s_{l+\alpha}.$$

这时定义

$$O(y)=\{x\in[0,\ f(c)]\mid I(x)\text{ 与 }I(y)\text{ 的第 }1,\ \cdots,l-1,l+1,\cdots,l+\alpha\text{ 位相同}\}.$$

它是一个开集. 对于 $x\in O(y)$，如果 $I(x)$ 的第 l 位为 0 或 c，则已成立 $I(x)<t$. 如果 $I(x)$ 的第 l 位为 1，则从 $t_1\cdots t_l$ 为奇串可见也成立 $I(x)<t$. 这样就证明了 $O(y)\subseteq L_t$.

再考虑 $t_{l+1}\cdots t_n=s_{l+1}\cdots s_n$ 的情况．这时定义

$$O(y)=\{x\in[0,f(c)]\mid I(x)\text{与}I(y)\text{的长为}n\text{的前缀除了第}l\text{位外相同}\}.$$

由于 $s_1\cdots s_n$ 中除了 $s_l=c$ 外不出现符号 c，$O(y)$ 仍为开集．对于 $x\in O(y)$，从反证法假设，$I(x)$ 的第 l 位不会是 1(否则 $I(x)$ 即以 t 为前缀)．这导致 $I(x)<t$ 成立，从而有 $O(y)\subseteq L_t$．对 $t_l=1$ 的讨论已完毕．对于 $t_l=0$ 的讨论从略．

现在讨论揉序列 KS 含符号 c 的情况．这时的临界点 c 为周期点．下面我们只指出与上面不同的几个地方，而不详细写出全部过程．

仍设 $y\in L_t$，$I(y)=s_1s_2\cdots s_n\cdots$．设已有

$$s_1\cdots s_{l-1}=t_1\cdots t_{l-1},\qquad s_l\neq t_l.$$

仍考虑 $t_l=1$，$s_l=c$ 和 $l<n$ 的情况．

如果 $s_{l+1}\cdots s_n$ 是 KS 的前缀，但其中不出现符号 c，则讨论与以前相同．

如果 $s_{l+1}\cdots s_n$ 中出现符号 c，包括出现多个符号 c 的情况，则可以先定义

$$O(y)=\{x\in[0,\ f(c)]\mid I(x)\text{与}I(y)\text{在长为}n\text{的前缀中除符号}c\text{所占的位置外完全相同}\}.$$

然后利用临界点 c 的有限次原象只有有限多个的事实，缩小 $O(y)$ 为

$$O_1(y)=\{x\in O(y)\mid\text{当}x\neq y\text{时在}I(x)\text{的长为}n\text{的前缀中不含}c\}.$$

$O_1(y)$ 为 y 的开邻域．

考虑 $u\in O_1(y)$，$x\neq y$．先看 $I(y)$ 与 t．设

$$I(y)=s_1\cdots s_{l-1}c\,s_{l+1}\cdots s_{l+a-1}c\,s_{l+a+1}\cdots,$$

在 $s_{l+1}\cdots s_{l+a-1}$ 中不出现 c．只需要讨论 $t_{l+1}\cdots t_{l+a-1}=s_{l+1}\cdots s_{l+a-1}$ 的情况．这时可写

$$t=s_1\cdots s_{l-1}1s_{l+1}\cdots s_{l+a-1}t_{l+a}\cdots.$$

现在记 $$I(x)=s_1\cdots s_{l-1}x_l s_{l+1}\cdots s_{l+\alpha-1}x_{l+\alpha}\cdots.$$
由于 $t_1\cdots t_l=s_1\cdots s_{l-1}1$ 为奇串，$t_{l+1}\cdots t_{l+\alpha}$ 为偶串. 如果 $x_l=0$, 则已有 $I(x)<t$ 成立. 如果 $x_l=1=t_l$, 则考察 $x_{l+\alpha}$. 从 $O_1(y)$ 的定义，可见 $x_{l+\alpha}$ 不是 0 就是 1. 如 $x_{l+\alpha}\neq t_{l+\alpha}$, 则从 $t_1\cdots t_{l+\alpha}$ 奇，也得到 $I(x)<t$. 如果 $x_{l+\alpha}=t_{l+\alpha}$, 则 $I(x)$ 与 t 的前 $l+\alpha$ 位完全一样，且为奇串. 注意到 $I(y)$ 从 $s_{l+\alpha+1}$ 开始重复 $s_{l+1}\cdots s_{l+\alpha-1}0\cdots$, 又已知 $s_{l+1}\cdots s_{l+\alpha-1}t_{l+\alpha}$ 为偶串，我们可以用同样的方法讨论下去. 按照反证法假设，$I(x)$ 不可能以 t 为前缀. 这时就会与上面一样推出 $I(x)<t$. 于是已建立 $O_1(y)\subseteq L_t$.

以上已证明 L_t 为非空开集. 对于 R_t 的讨论相同. 定理 1 证明完毕.

注：以上所用的方法与[22] 中证明 § 5.4 的充分条件部分的方法是相同的. 但实际上这里是用反证法来进行的. 在证明了
$$\{x\in[0,\ f(c)]\mid I(x)\text{以 } t \text{ 为前缀}\}\neq\varnothing$$
之后，可以看出它是开集. 从而 L_t 与 R_t 都是闭集.

A.2 定理 3 的证明

定理 3 如果揉序列 KS 不是周期序列，则有
$$\mathrm{L}_1=\mathrm{L}=\mathscr{L}(\mathrm{KS}).$$
这里语言
$$\mathrm{L}=\{t\in\{0,\ 1\}^*\mid \text{存在无限串 } s, \text{以 } t \text{ 为前缀，且对每个 } i\geqslant 0 \text{ 有 } \sigma^i(s)\leqslant \mathrm{KS}\},$$
$$\mathrm{L}_1=\{t\in\{0,\ 1\}^*\mid \text{存在无限串 } s, \text{以 } t \text{ 为前缀，且对每个 } i\geqslant 0 \text{ 有 } \sigma^i(s)<\mathrm{KS}\}$$
(参看 § 6.1 与 § 6.5).

证明 从定义已有 $\mathrm{L}_1\subseteq\mathrm{L}$. 只要证 $\mathrm{L}\subseteq\mathrm{L}_1$.

由于 L 和 L_1 完全由揉序列 KS 所确定，而与以 KS 为揉序列的具体映射无关，因此可以对某一个这样的映射来证明 $\mathrm{L}=\mathrm{L}_1$.

利用二次方映射当参数从 0 到 4 时为完全族，即能取到任何

一个单峰映射的揉序列(参见[22]),因此存在 b_0, 使映射 $f(x)=b_0x(1-x)$以定理中的 KS 为其揉序列.

现设 $t\in L$. 根据 A.1 中的定理 1, 存在点 $x\in[0, f(c)]$, 使踪迹 $I(x)$以 t 为前缀. 这样的点 x 全体所成的集是开的. 利用临界点 c 的所有原象不超过可列个,我们可以取出一个点 x, 使 $I(x)$ 不仅以 t 为前缀,而且不含符号 c. 我们还不妨设已有 $x<f(c)$.

从必要条件(本书第44页中之基本结论 1)可知,对每个 $i\geqslant1$, 成立 $\sigma^i(I(x))\leqslant\mathrm{KS}$ (由于 $x\leqslant f(c)$, 在 $i=0$ 时也成立).

我们要证明,对这个踪迹 $I(x)$来说,于每个 $i\geqslant0$ 时成立

$$\sigma^i(I(x))<\mathrm{KS}.$$

在 $i=0$ 时从 $x<f(c)$可见已不成问题. 然后,由于 t 是 $I(x)$的前缀,就得到所要的结论 $t\in L_1$.

用反证法,如对某个 $i>0$, 有

$$\sigma^i(I(x))=\mathrm{KS}(=I(f(c))).$$

利用

$$\sigma^i(I(x))=I(f^i(x)),$$

可见 $f^i(x)$与$f(c)$的踪迹完全相同,因此区间 $[f^i(x), f(c)]$ 中每个点的踪迹都等于 KS, 即所谓同宿区间(Homterval). 由于这时的 KS 为非周期,而二次方映射满足施瓦兹导数为负的条件,因此从[22]中的有关定理知道,映射 $f(x)=b_0x(1-x)$不存在稳定周期轨,也不存在非退化的同宿区间,于是只能成立

$$f^i(x)=f(c).$$

但由于这时已有 $i>0$, 而 $f(c)$的原象只可能是临界点 c, 因此 $f^{i-1}(x)=c$. 这与踪迹 $I(x)$中不含符号 c 相矛盾. 证毕.

注: 如果揉序列是含符号 c 的周期序列, 那么定理 3 的结论是平凡的. 但在揉序列为不含符号 c 的周期序列时, 容易举例说明

$$L\subsetneq L_1$$

是可能的.

附录 B

$\mathscr{L}$(KS)为正规语言的充分条件

在这个附录中证明在§8中的定理2和定理3(见§8.4)，即当揉序列 KS 为周期序列和终极周期序列时，$\mathscr{L}$(KS) 为正规语言。

此外，在 B.3 中引进既约串及有关事实，在 B.5 中介绍循环移位最大字的概念，对于本书的有关部分都是不可缺少的工具。

B.1 关于周期揉序列的一个引理

在§7.5介绍了关于 $z\in\mathscr{L}$(KS)的一般判定法则．对于周期揉序列，可以有更为方便的结果．

引理 设 $\mathrm{KS}=x^\infty$，x 为偶串，长度 $|x|=n$，那么 $z\in\mathscr{L}(\mathrm{KS})$ 的充分必要条件是，z 的每一个长度不超过 n 的子串 z' 满足条件

$$z'\leqslant x.$$

证明 由于 $z'\leqslant x$ 与 $z'\leqslant x^\infty(=\mathrm{KS})$ 等价，而当

$$z\in \mathrm{L}=\mathscr{L}(\mathrm{KS})$$

时，也有 $z'\in \mathrm{L}$，因此必要性不成问题．以下讨论充分性．

用反证法．设有符号串 $z\in S^*$，$z\notin \mathrm{L}$，但 z 的每一个长度不超过 n 的子串 z' 都满足条件 $z'\leqslant x$．当然，只需讨论 $|z|>n$ 的情况．

应用§7.5的判定法则，从 $z\notin \mathrm{L}$ 可见 z 有后缀 $v>\mathrm{KS}=x^\infty$．根据反证法条件，当然有 $|v|>n$ 成立．现在观察 v 的长为 n 的前缀．从 $v>\mathrm{KS}=x^\infty$ 可见它不能小于 x．另一方面，从条件已知它不能大于 x．因此可见 v 以 x 为前缀．

利用 x 为偶串，可继续讨论 v 的长度为 $n(=|x|)$ 倍数的前缀，得到分解式

$$v=x^iv',$$

其中 $l>0$, $|v'|<n$. 由于 x 为偶串, 易见

$$v>\mathrm{KS} \Leftrightarrow v'>\mathrm{KS} \Leftrightarrow v'>x.$$

但 v' 也是 z 的一个子串, 长度小于 n, 因此与引理条件相矛盾. 证毕.

注: 如 $\mathrm{KS}=x^\infty$ 中 x 为奇串, 则应当用 x^2 代替 x 后再应用上述引理.

B.2 定理 2 的证明

将 § 8.4 中的定理 2 重复如下.

定理 2 如 KS 为周期序列, 则 $\mathscr{L}(\mathrm{KS})$ 为正规语言.

证明 应用在 § 6.2 中的讨论, 不妨假定揉序列中不含符号 C, 记为 $\mathrm{KS}=x^\infty$. 同时设 x 已为偶串.

主要关键是证明有等价关系 $xR_{\mathrm{L}}x^2$ 成立, 然后应用 § 8.7 中的定理 4, 就可以得到所要的结论. 这里 R_{L} 是由语言 $\mathrm{L}=\mathscr{L}(x^\infty)$ 在 S^* 中引入的自然等价关系(参见 § 1.8 与 § 8、§ 9).

任取 $z\in S^*$. 从 $x^2z\in \mathrm{L}\Rightarrow xz\in \mathrm{L}$ 是显然成立的, 因为 xz 作为 x^2z 的子串当然属于 L(参见 § 7.4 性质 1). 从 $xz\in L\Rightarrow x^2z\in \mathrm{L}$ 可讨论如下. 如果 $xz\in \mathrm{L}$, 则应用 B.1 的引理, 可见串 xz 的任何一个长度不超过 $|x|(=n)$的子串都 $\leqslant x$. 但这立即推出 x^2z 也具有此性质. 再次应用引理, 就得出 $x^2z\in \mathrm{L}$. 这样就证明了所要求的等价关系. 证毕.

注: 在[35]、[36]中, 通过构造接受 $\mathrm{L}=\mathscr{L}(x^\infty)$的有限自动机来证明定理 2(参看本书 § 8.5 与 § 8.6 等).

B.3 关于既约串的基本概念和事实

在定理 3 的证明之前需要介绍有关既约符号串的知识. 实际上, 在符号序列的研究中这是必不可少的.

定义 对于非空符号串 y, 如果它不能写成 $y=u^l$, 其中 u 为某个符号串, 指数 $l>1$, 则称 y 为既约(符号)串; 否则, 称 y 为非

既约(符号)串．又如 $y=u^l$，u 为既约串，则称 u 为 y 的原根 (Primitive Root)(参见[4])．

这方面的最基本事实有

引理 一个非空串 y 为非既约串的充分必要条件是存在两个非空串 y_1 和 y_2，使

$$y=y_1y_2=y_2y_1.$$

证明 必要性是明显的，如 $y=u^l$，$l>1$，则取 y_1 和 y_2 为 u 的幂即可．

反之，如果存在上述两个非空串 y_1 和 y_2，使 $y=y_1y_2=y_2y_1$，则考察串 y_1 和 y_2 的长度．如果 $|y_1|=|y_2|$，则也有 $y_2=y_1$．因此令 $u=y_1(=y_2)$，即有 $y=u^2$ 成立．对于两者长度不等的情况，不妨设 $|y_1|>|y_2|$．从 $y_1y_2=y_2y_1$ 可看出此时 y_1 以 y_2 为前缀．记 $y_1=y_2^my_3$，$m\geqslant 1$，且 $|y_3|<|y_2|$，则有

$$y_1y_2=y_2^my_3y_2=y_2y_1=y_2^{m+1}y_3\Rightarrow y_3y_2=y_2y_3,$$

继续做下去，恰如求两个自然数的最大公因子时的辗转相除法那样，可以得到 u，使 $y=u^l$，$l>1$．证毕．

注：这时 y_1 和 y_2 也是 u 的幂，这是一个有用的事实．

B.4 定理 3 的证明

将 § 8.4 中的定理 3 重复如下．

定理 3 如 $\mathrm{KS}=\rho\lambda^\infty$，则 $\mathscr{L}(\mathrm{KS})$ 为正规语言．

证明 不妨设在 $\rho\lambda^\infty$ 中的符号串 λ 已为既约符号串．对于 λ 为偶既约串情况，令

$$k=[|\rho|/|\lambda|]+2,$$

我们将要证明 KS 的两个前缀为 R_L 等价：

$$\rho\lambda^kR_\mathrm{L}\rho\lambda^{k+1}.$$

对于 λ 为奇既约串的情况，对同样的 k，可以证明

$$\rho\lambda^kR_\mathrm{L}\rho\lambda^{k+2}.$$

然后，只要应用 § 8.7 中的定理 4 就可以得到所要求的结果．由于

上述两种情况的讨论大同小异，我们在下面只给出 λ 为偶既约串时的证明.

任取 $z\in S^*$，我们来证明

$$\rho\lambda^k z\in \mathrm{L}\Rightarrow \rho\lambda^{k+1}z\in \mathrm{L}.$$

如果 z 是 λ^∞ 的前缀，则显然成立. 否则，可以分解 $z=z'z''$，其中 z' 的任何真前缀是 λ^∞ 的前缀，但 z' 本身则不是. 由此可见，将 z' 的最后一个符号取反码后得到的串 $\hat{z}'$ 则是 λ^∞ 的前缀. 因此可写出

$$z=z'z'',\qquad z'=\lambda^{k'}\hat{\lambda}',$$

其中 $k'\geqslant 0$, λ' 是 λ 的一个前缀. 这时有

$$\rho\lambda^k z\in \mathrm{L}\Rightarrow \rho\lambda^k z'\in \mathrm{L}.$$

记

$$\rho\lambda^k z'=\rho\lambda^{k+k'}\hat{\lambda}',$$

然后应用 § 7.6，记它关于 KS 的最长前后缀为 u. 因为 $\rho\lambda^k z'$ 本身不是 KS 的前缀，所以

$$|u|<|\rho\lambda^k z'|.$$

如果 $|u|>|\rho\lambda|$，则因为 u 是 KS 的前缀，可记为 $u=\rho\lambda^l\lambda''$，$l>0$, λ'' 为 λ 的真前缀. 但同时 u 又是 $\rho\lambda^k z'=\rho\lambda^{k+k'}\hat{\lambda}'$ 的后缀. 利用 λ 为既约串和 B.3 的引理，只能导致 $\lambda''=\hat{\lambda}'$. 但 λ'' 为 λ 的前缀，而 $\hat{\lambda}'$ 则不是，引出矛盾.

现设 $|u|\leqslant|\rho\lambda|$. 从 k 的选择可看出，u 既是 $\rho\lambda^k z'$ 的后缀，则也是 $\lambda^k z'$ 的后缀. 由此又推出 u 也是 $\rho\lambda^{k+1}z'$ 的最长前后缀. 应用 § 8.7 定理 4 的证明中的第一个结果，可见成立

$$\rho\lambda^k z' R_{\mathrm{L}}\rho\lambda^{k+1}z',$$

因为它们关于 KS 的最长前后缀相同.

再利用等价关系 R_{L} 的右不变性质(参见 § 1.8)，从 $z=z'z''$，即可导出

$$\rho\lambda^k z R_{\mathrm{L}}\rho\lambda^{k+1}z.$$

因此已建立了

$$\rho\lambda^k z\in \mathrm{L}\Rightarrow \rho\lambda^{k+1}z\in \mathrm{L}.$$

相反关系的建立过程完全类似，从略. 这样就证明了所要的结论.

注：在[37]中的证明方法与此不同(参看本书§8.5、§8.6和§9).

B.5 循环移位最大字

现在引进循环移位最大字的概念，且利用既约性导出一个有用的事实.

设 $x=x_1x_2\cdots x_n$ 是长度为 n 的有限串. 定义

$$\sigma(x)=x_2x_3\cdots x_nx_1$$

为 x 的一次循环移位. 一般地有

$$\sigma^i(x)=x_{i+1}\cdots x_nx_1\cdots x_i, \qquad 1\leqslant i<n$$

和 $\sigma^n(x)=x$.

这里对于循环移位的记号与§5.9中的移位算子 σ 的记号相同，它在具体场合的含义可以由上下文来区分开.

定义 $M(x)$是在 $\{\sigma^i(x)\}_{1\leqslant i\leqslant n}$ 中按序为最大的一个，并称 $M(x)$为串 x 的循环移位最大字(串). 当然，在 $\sigma(x)$、$\sigma^2(x)$、…、$\sigma^n(x)(=x)$中可能有一个以上的串等于 $M(x)$.

如果有 $x=M(x)$，则称 x 为循环移位最大字(串).

对于 $KS=x^\infty$，因为 KS 为移位最大字，所以必有 $x=M(x)$. 反之，如果有 $x=M(x)$，则 x^∞ 为移位最大字，即可以是某个单峰映射的揉序列.

利用 B.1 的引理可见，如果 x 为既约串，则 $\{\sigma^i(x)\}_{1\leqslant i\leqslant n}$ 中任何两个都不相同，因此只有唯一的一个 $i\in\{1, \cdots, n\}$，使 $\sigma^i(x)=M(x)$成立.

附录 C

关于 § 10.4 的补充

在本附录中给出在 § 10.4 中的命题的证明，然后介绍它的推广及其应用.

C.1 命题的证明

重新叙述在 § 10.4 中的命题如下.

命题 如果 s 是费根鲍姆吸引子的允许字，$KS=t_\infty=\lim_{k\to\infty} t_k$，同时 s 以某个 $t_k(k\geqslant 0)$ 为前缀，即有 $s=t_kv$，那么 v 的长度为 $2^k(=|t_k|)$ 的前缀只有两种可能：1° v 以 t_k 为前缀；2° v 以 $\hat{t}_k$ 为前缀.

证明 对 k 用数学归纳法. 在 $k=0$ 时 $t_0=1$，因此是平凡的. 现设上述论断对 k 以及比 k 小的自然数已成立，讨论 $k+1$ 时的情况.

将允许字 s 写成

$$s=t_{k+1}v=t_k\hat{t}_kv=t_{k-1}\hat{t}_{k-1}t_{k-1}t_{k-1}v,$$

对其后缀 $t_{k-1}v$ 用归纳假设(s 为允许字时 $t_{k-1}v$ 也是允许字)，可见 v 的长为 2^{k-1} 的前缀不是 t_{k-1}，就是 $\hat{t}_{k-1}$. 但从

$$t_{k+1}\hat{t}_{k-1}>t_{k+1}t_{k-1}\cdots=t_\infty,$$

可见 v 不能以 $\hat{t}_{k-1}$ 为前缀. 于是有 $v=t_{k-1}v'$.

这时 $s=t_{k+1}t_{k-1}v'$. 对 $t_{k-1}v'$ 用归纳假设，可见 v' 以 t_{k-1} 或 $\hat{t}_{k-1}$ 为其前缀. 又从

$$t_{k+1}t_{k-1}t_{k-1}>t_{k+1}t_{k-1}\hat{t}_{k-1}\cdots=t_\infty,$$

可见 v' 只能以 $\hat{t}_{k-1}$ 为前缀. 记 $v'=\hat{t}_{k-1}v''$，则有

$$s=t_{k+1}t_{k-1}\hat{t}_{k-1}v''=t_{k+1}t_kv''$$

再对 $t_k v''$ 用归纳假设，这时 v'' 的长为 2^k 的前缀不是 t_k 就是 $\hat{t}_k$. 对前者，$s=t_{k+1}t_k^2\cdots=t_{k+1}\hat{t}_{k+1}$; 对后者，$s=t_{k+1}t_k\hat{t}_k=t_{k+1}t_{k+1}\cdots$, 已证明完毕.

C.2 推　广

上述命题可以推广成相当一般的引理，它不仅可推出上述命题，而且包含了在文献中的一些其他结果为其特例. 这里的问题是，一个符号串的某个前缀，在什么条件下对符号串的其余部分起某种制约作用.

引理　设 t 和 s 是只由符号 0 和 1 组成的串，或均为有限串，或均为无限串. 如果有

$$t=uv,\quad s=uw,\qquad s\leqslant t,\quad w\leqslant v,$$

同时 u 为奇串，那么 $v=w$(即 $t=s$).

证明　不妨讨论无限串情况. 记

$$v=v_1v_2\cdots v_n\cdots,\qquad w=w_1w_2\cdots w_n\cdots.$$

用数学归纳法证明对每个 $n\geqslant 1$ 成立 $v_n=w_n$.

在 $n=1$ 时，如 $v_1=1$，则从 $s\leqslant t$ 和 u 为奇串，可见 $w_1=1$. 如 $v_1=0$，则从 $w\leqslant v$ 得出 $w_1=0$.

现设在 $1\leqslant i\leqslant n$ 时已成立 $v_i=w_i$. 讨论 v_{n+1} 和 w_{n+1}. 如果 $v_1\cdots v_n$ 是偶串，$v_{n+1}=1$，则从 $s\leqslant t$ 和 $uv_1\cdots v_n$ 是奇串，可见 $w_{n+1}=1$. 如 $v_{n+1}=0$，则从 $w\leqslant v$ 得出 $w_{n+1}=0$.

在 $v_1\cdots v_n$ 是奇串时，如 $v_{n+1}=1$，则从 $w\leqslant v$ 可见 $w_{n+1}=1$. 如 $v_{n+1}=0$，则从 $uv_1\cdots v_n$ 为偶串，$s\leqslant t$，就得到 $w_{n+1}=0$. 证毕.

C.3 从奇串平方开始的移位最大字

在符号动力学讨论中一个很有用的结果可以从上述引理推出.

推论　如 t 为移位最大字，且以 y^2 为前缀，y 为奇串，那么 $t=y^\infty$.

证明　记

$$t=yyu\cdots,\qquad s=\sigma^{|y|}(t)=yu\cdots,$$

其中 $|u|=|y|$. 由于 t 为移位最大字，有

$$s\leqslant t,\qquad \sigma^{2|y|}(t)=u\cdots\leqslant t=y\cdots,$$

即 $u\leqslant y$. 应用 C.2 的引理，可见 $u=y$. 于是得到 $t=y^3\cdots$. 一般地，设已知 $t=y^n u\cdots$, $n\geqslant 2$, 就可以与刚才一样证明 $v=y$, 从而 t 以 y^{n+1} 为前缀. 证毕.

C.4　其他例子

[例 1]　用 C.2 中的引理可以对 C.1 给出一个新的证明.

写出

$$t_\infty=t_k\hat{t}_k\cdots,\qquad s=t_k v,$$

并写 $v=v'v''$, 其中 v' 的长度为 $|t_k|-1=2^k-1$. 这时有 $s\leqslant t_\infty$ 和 $v\leqslant t_k$ 成立. 由于 t_k 与 $\hat{t}_k$ 只在最后一个符号上不同，因此可以应用引理，知道 v' 是 t_k (或 $\hat{t}_k$)的前缀. 于是根据 v'' 从 0 还是 1 开始，就可以推出 v 以 t_k 或 $\hat{t}_k$ 为前缀. 这就是 C.1 中命题的结论.

[例 2]　如 $\mathrm{KS}=y^\infty$, y 为奇串，s 是以 y 为前缀的允许字，那么 $s=y^\infty$.

实际上，写 $s=yu$, 则有

$$s\leqslant yy^\infty(=\mathrm{KS}),\qquad u=\sigma^{|y|}(s)\leqslant y^\infty.$$

应用引理，即得到 $u=y^\infty$.

附录 D

联系 $N(t)$ 与 $D(t)$ 的公式

在这里我们证明在 § 13.5 中的基本公式

$$N(t)=\frac{1}{2(1-t)}+\frac{1}{2(1-t)D(t)},\tag{1}$$

其中 $N(t)$是 $L=\mathscr{L}(KS)$的生成函数，其定义为

$$N(t)=1+\sum_{n=1}^{\infty}N_n t^n.\tag{2}$$

$D(t)$是由揉序列 $KS=a_1a_2\cdots a_n\cdots$确定的揉行列式，在 KS 不含符号 c 时，$D(t)$的定义为

$$D(t)=1+\varepsilon_1 t+\varepsilon_1\varepsilon_2 t^2+\cdots+\varepsilon_1\cdots\varepsilon_n t^n+\cdots,\tag{3}$$

这里 $\varepsilon_i=\varepsilon(a_i)$，函数 ε 的定义为

$$\varepsilon(0)=1,\qquad \varepsilon(1)=-1.$$

在 KS 含 c 时，可以按 § 6.2 中的方法处理，或按[21]中的方法写出 $D(t)$，两者完全一致.

利用在 § 13.3 中引入的序列 $\{S_n\}_{n\geq 1}$，可以定义函数

$$S(t)=\frac{1}{2}+\sum_{n=1}^{\infty}S_n t^n.\tag{4}$$

我们先证明联系 $S(t)$ 与 $D(t)$的公式

$$S(t)=\frac{1}{2(1-t)D(t)},\tag{5}$$

然后只要用 $S_n=N_n-N_{n-1}(n\geqslant 1)$ 即可导出公式(1).

公式(5)等价于对 $n\geqslant 2$ 的恒等式组

$$S_n=-\varepsilon_1 S_{n-1}-\varepsilon_1\varepsilon_2 S_{n-2}-\cdots-\varepsilon_1\cdots\varepsilon_{n-1}S_1+\frac{1}{2}(1-\varepsilon_1\cdots\varepsilon_n).\tag{6}$$

以下设 KS 从符号 1 开始，即 $a_1=1$. 这时总成立 $S_1=1$ (实际上，

KS 如从符号 0 开始，则只能是 0^{ω})．

现在直接证明：对任何 $n \geqslant 2$，恒等式(6)成立．

按照定义，S_n 是语言 $\mathscr{L}(\mathrm{KS})$ 中长为 n 且从符号 1 开始的字的个数．我们可以将(6)的右方的每一项与一种计数操作等同起来，并且证明这些计数操作的总和恰恰是 S_n．

右方第一项的计数意义是清楚的．因为 $a_1=1$、$\varepsilon_1=\varepsilon(a_1)=-1$，所以，可以将 $-\varepsilon_1 S_{n-1}$ 看成是对于语言 $\mathscr{L}(\mathrm{KS})$ 中长度 n 且从 11 开始的字的计数(参见 § 13.3 的例 1)．

但从第二项开始，事情没有这样简单．我们不妨设想每一项的计数操作都记录在一个帐本上．在系数 $-\varepsilon_1\cdots\varepsilon_i>0$ 时，将(6)中右方对应的项 $-\varepsilon_1\cdots\varepsilon_i S_{n-i}$ 看成向帐本中加入数量为 S_{n-i} 的符号串，长度均为 n，形状均为 $a_1a_2\cdots a_iy$，其中 $y\in\mathscr{L}(\mathrm{KS})$，而且均从符号 1 开始．这时 y 的长度为 $n-i$．从 $\varepsilon_1\cdots\varepsilon_i=-1$ 可知 KS 的前缀 $a_1\cdots a_i$ 是奇串．当然，这里并未保证这些串一定在语言 $\mathscr{L}(\mathrm{KS})$中．

在 $a_1\cdots a_i$ 为偶串时 $-\varepsilon_1\cdots\varepsilon_i=-1$，我们将(6)右方相应的项解释为从帐本中删去上述形式的 S_{n-i} 个串．

我们从(6)右方第一项起逐项进行上述计数操作．这里允许所谓欠帐操作，即是要删去的串尚未入帐时先要删除也是允许的，可以先记下来，然后在今后能"归还"时算清．这样做下去直到最后一项．当 $\varepsilon_1\cdots\varepsilon_n=-1$ 时规定将揉序列的前缀 $a_1\cdots a_n$ 计入帐本，它对应于最后一项 $(1-\varepsilon_1\cdots\varepsilon_n)/2$ 为 1 的情况，而在 $\varepsilon_1\cdots\varepsilon_n=1$ 时则不进行任何计数．

对于一个长度为 n 的符号串 z，它有可能记入帐本或从帐本中删去，并且有可能多次出现这两种情况．我们可以将记入帐本的动作对应于 $+1$，而将从帐本中删去的动作对应于 -1．

我们将要证明，如 $z\in\mathscr{L}(\mathrm{KS})$，且是从符号 1 开始的长为 n 的串，则在计数活动全过程中按上述 ±1 规定所得的代数和恰为 $+1$．反之，如 $z\notin\mathscr{L}(\mathrm{KS})$，且为从 1 开始的长为 n 的串，则这个代

数和必是 0，其中包括从不参加计数活动的串在内．容易理解，只要这两个论断成立，则公式(6)就成立．然后，公式(5)与(1)也都成立．

现在开始证明以上论断．

对于(6)右方第一项所计入的 $z\in\mathscr{L}(\mathrm{KS})$，都是从 11 开始且长为 n 的串．按上述规定不会参与今后的任何计数操作，因此代数和即为 $+1$．这是最为简单的情况．

现设操序列的前缀为

$$a_1a_2\cdots a_n=10^{n_1}10^{n_2}1\cdots10^{n_k},$$

其中 $n_i\geqslant0$，$i=1,\cdots,k$，$\sum\limits_{i=1}^{k}n_i+k=n$．设 z 是长度为 n 且从符号 1 开始的串．分以下几种情况讨论．

1．$z\in\mathrm{L}=\mathscr{L}(\mathrm{KS})$，但不以 10^{n_1} 为前缀．

这时 z 将以

$$11,\ 101,\ \cdots,\ 10^{n_1-1}1$$

中的某一个为其前缀．因 $z\in\mathrm{L}$，故在与 S_{n-1}、…、S_{n-n_1} 对应的 n_1 步操作中必将这样的 z 记入帐本，而且不会再删去．反之，又可看出，在这 n_1 步中入帐的每一个串都属于 L（应用 §7.5）．注意上述各项系数均为 $+1$．

2．$z\in\mathrm{L}$，以 10^{n_1} 为前缀，但 $z\neq a_1\cdots a_n$．

这时可设 z 的形式为

$$z=10^{n_1}1\cdots10^{n_l}y,\qquad 1\leqslant l\leqslant k-1,$$

其中 y 不以 $10^{n_{l+1}}$ 为前缀．于是 y 以

$$0,\ 11,\ 101,\ \cdots,\ 10^{n_{l+1}-1}$$

中的某一个为其前缀．

由于 $z\in\mathrm{L}$，$y<10^{n_{l+1}}$，可见 l 是偶数．这时 $10^{n_1}1\cdots10^{n_l}y\in\mathrm{L}$，因此在与 10^{n_1} 对应的计数操作时已将 z 入帐了．但当 y 不是从 0 开始时，在与 $10^{n_1}10^{n_2}$ 对应的计数操作中又会将 y 从帐本中删去．由于 l 是偶数，这两种操作交替进行，直到最后一次在与 $10^{n_1}1\cdots$

$10^{n_l}10^p$ 对应的计数操作中将 z 计入帐内，并不再变化. 这里 p 是串 $y=10^p1\cdots$ 中的串 0^p 的长度. 当 y 从 0 开始时，计数操作少两次，仍为奇数，代数和为 $+1$.

3. $z\notin \mathrm{L}$. 如 $z=10^py, p<n_1, y$ 从符号 1 开始，则从 §7.5 可见

$$10^py\notin \mathrm{L} \Leftrightarrow y\notin \mathrm{L},$$

因此这样的串在计数全过程中都不会参与进去.

现设 $z=10^{n_1}\cdots10^{n_l}y$，$y$ 不以 $10^{n_{l+1}}$ 为前缀，且 $y\in \mathrm{L}$. 如存在 i，$1\leqslant i<l$，使 $10^{n_{i+1}}1\cdots10^{n_l}y\in \mathrm{L}$，则取 i 为最小可能的数. 应用 §7.5，可看出

$$10^{n_i}1\cdots10^{n_l}y>a_1\cdots a_n,$$

又从移位最大字的性质有

$$a_1\cdots a_n>10^{n_i}1\cdots10^{n_l}10^{n_{l+1}}$$

和 $y=10^{n_{l+1}}$，可推出 $10^{n_i}1\cdots10^{n_l}$ 是奇串. 因此 i 与 l 的奇偶性相同. 同理可见 $i<l-1$ 成立.

讨论 i 为偶数的情况. 从 $10^{n_1}1\cdots10^{n_i}$ 为偶串，在这时将串 z 从帐本中删去(即欠帐操作). 在这之后入帐与删去操作交替进行，直到最后一次删去为止. 对于 i 为奇数的情况是类似的.

对于上述 i 不存在时的讨论，与上面的第 2 种情况相仿.

总之，在这里的入帐与删除的代数和为零，因此在 $z\notin \mathrm{L}$ 时最终不进入帐本.

4. $z=a_1a_2\cdots a_n(=10^{n_1}1\cdots10^{n_k})$.

如 k 为偶数，则 $\varepsilon_1\cdots\varepsilon_n=1$. 这个串在与 $10^{n_1}1\cdots10^{n_{k-1}}$ 所对应的入帐操作中已进入帐本，并不会再变动. 如 k 为奇数，则上述最后一次操作是删去. 按照(6)的最后一项

$$(1-\varepsilon_1\cdots\varepsilon_n)/2=1,$$

按规定将它加入. 证毕.

注：从公式(5)可见 $S(t)$ 的收敛半径与 $N(t)$ 相同，也是 $1/s$. 因此，从数列 $\{S_n\}_{n\geq1}$ 也可以直接计算语言 $\mathrm{L}=\mathscr{L}(\mathrm{KS})$ 的增长数 s 与熵 h.

参 考 文 献

[1] J. Hopcroft and J. Ullman, *Introduction to Automata Theory, Languages and Computation*, Addison-Wesley, Reading, Mass., 1979.

[2] A. Salomaa, *Formal Languages*, Academic Press, New York, 1973.

[3] M. Minsky, *Computation: Finite and Infinite Machines*, Englewood Cliffs, N. J., Prentice Hall, 1967.

[4] A. Salomaa, *Jewels of Formal Language Theory*, Computer Science Press, Rockville, 1981.

[5] A. Salomaa, *Computation and Automata*, Cambridge Univ. Press, Cambridge, 1985.

[6] G. E. Révész, *Introduction to Formal Languages*, McGraw-Hill Book Company, New York, 1983.

[7] A. M. Turing, *On computable numbers with an application to the Entscheidungs problem*, Proc. London Math. Soc. **2**(1936)544.

[8] N. Chomsky, *Three models for the description of language*, IRE Trans. IT-2(1956) 113.

[9] N. Chomsky, *On certain formal properties of grammars*, Information and Control **2**(1959) 137.

[10] S. Ginsburg, *The Mathematical Theory of Context-Free Languages*, McGraw-Hill, New York, 1966.

[11] W. Ogden, *A helpful result for proving inherent ambiguity*, Math. Systems Theory **2**(1968) 191.

[12] G. T. Herman, G. Rozenberg, *Developmental Systems and Languages*, North-Holland Publ., Amsterdam, 1975.

[13] G. Rozenberg, A. Salomaa, *The Mathematical Theory of L Systems*, Academic Press, New York, 1980.

[14] P. Prusinkiewicz, J. Hanan, *Lindenmayer Systems, Fractals, and*

Plants, Springer-Verlag, New York, 1989.

[15] P. Prusinkiewicz, A. Lindenmayer, *The Algorithmic Beauty of Plants*, Springer-Verlag, New York, 1990.

[16] A. V. Aho, *Indexed grammars-an extension of context-free grammars*, J. ACM **15** (1968) 647.

[17] K. Čulik II, *On some families of languages related to developmental systems*, Intern. J. Computer Math. **4**(1974) 31.

[18] Li T. Y., J. A. Yorke, *Period three implies chaos*, Am. Math. Monthly **82**(1975) 985.

[19] A. N. Sharkovskii, *Coexistence of the cycles of a continuous mapping of the line into itself*, Ukranian Math. Z. **16**(1964) 61.

[20] R. M. May, *Simple mathematical models with very complicated dynamics*, Nature **261**(1976) 459.

[21] J. Milnor, W. Thurston, *On iterated maps of the interval*, Lecture Notes in Math. **1342**(1988) 465.

[22] P. Collet, J. -P. Eckmann, *Iterated Maps on the Interval as Dynamical Systems*, Birkhäuser, Boston, 1980.

[23] C. Preston, *Iterates of Maps on an Interval*, Lecture Notes in Math. 999, Springerverlag, New York, 1983.

[24] R. L. Devaney, *An Introduction to Chaotic Dynamical Systems*, Benjamin, Menlo Park, 1986. 2nd Edition, Addison-Wesley, Redwood City, 1989.

[25] C. Mira, *Chaotic Dynamics*, World Scientific, Singapore, 1987.

[26] Hao B.-L., *Elementary Symbolic Dynamics and Chaos in Dissipative Systems*, World Scientific, Singapore, 1989.

[27] W. de Melo, *Lectures on One-Dimensional Dynamics*, IMPA, CNPq, 1990.

[28] 张景中,熊金城,函数迭代与一维动力系统.四川教育出版社,1992.

[29] L. S. Block, W. A. Coppel, *Dynamics in One Dimension*, Lecture Notes in Math. **1513**, Springer-Verlag, New York, 1992.

[30] M. Morse, *A one-to-one representation of geodesics on a surface of negative curvature*, Am. J. Math., **43** (1921) 33.

[31] M. Morse, *Recurrent geodesics on a surface of negative curvature*, Trans. AMS. **22**(1921) 84.

[32] M. Morse, G. A. Hedlund, *Symbolic dynamics*, Am. J. Math., **60** (1938) 815.

[33] M. Morse, G. A. Hedlund, *Symbolic dynamics II*, Am. J. Math **62**(1940) 1.

[34] W. H. Gottschalk, G. A. Hedlund, *Topological Dynamics*, AMS. Colloquium Publ. **36**, Providence, 1955.

[35] Chen X., Lu Q.-H., Xie H.-M., *Grammatical complexity of one-dimensional dynamical systems*, Suzhou University, CNS Preprint Series # 009(1992).

[36] 陈曦,卢钦和,谢惠民,一维动力系统的语法复杂性,苏州大学学报(自然科学版) 9(1993) 68.

[37] Xie H.-M., *The finite automata of eventually periodic unimodal maps on the interval*, J. Suzhou Univ. **9**(1993) 112.

[38] Xie H.-M., *On formal languages of one-dimensional dynamical systems*, Nonlinearity **6** (1993)997.

[39] 王益,谢惠民,终极周期揉序列的最小有限自动机,(1993)[待发表].

[40] P. Grassberger, *Toward a quantitative theory of self-generated complexity*, Intern. J. Theor. Phys. **25**(1986) 907.

[41] P. Grassberger, *On symbolic dynamics of one-humped maps of the interval*, Z. Naturforsch. **43 a**(1988) 671.

[42] P. Grassberger, *Complexity and forecasting in dynamical systems*, in Measures of Complexity, Lecture Notes in Phys. 314, ed. L. Peliti and A. Vulpiani, Springer-Verlag, 1988, 1.

[43] J. P. Crutchfield, K. Young, *Inferring statistical complexity*, Phys. Rev. Letters **63**(1989) 105.

[44] J. P. Crutchfield, K. Young, *Computation at the onset of chaos, in Complexity, Entropy and the Physics of Information*, SFI Studies in the Sciences of Complexity, Vol. VIII, ed. W. H. Zurek, Addison-Wesley, 1990, 223.

[45] D. Auerbach, I. Procaccia, *Grammatical complexity of strange sets*, Phys. Rev. **A 41**(1990) 6602.

[46] D. Auerbach, *Dynamical complexity of strange sets, in Measures of Complexity and Chaos*, ed. N. B. Abraham, A. M. Albano, A. Passamante, P. E. Rapp, Plenum, 1990, 203

[47] E. J. Friedman, *Structure and uncomputability in one-dimensional maps*, Complex Systems **5**(1991) 335.

[48] C. Moore, *Generalized one-sided shifts and maps of the interval*, Norlinearity **4**(1991) 727.

[49] Hao B.-L., *Symbolic dynamics and characterization of complexity*, Physica **D 51**(1991) 161.

[50] J. Milnor, *On the concept of attractor*, Commun. Math. Phys. **99** (1985) 177.

[51] U. Kirchgraber, D. Stoffer, *On the definition of chaos*, Z. Angew, Math. Mech. **69**(1989) 175.

[52] L. Block, J. Guckenheimer, M. Misiurewicz, L.-S. Young, *Periodic points and topological entropy of one-dimensional maps*, Lecture Notes in Math. **819**, 1990, 18.

[53] S. Wolfram, *Computation theory of cellular automata*, Commun. Math. Phys. **95**(1984) 15.

[54] Chen X., Lu Q.-H., Xie H.-M. *Grammatical complexity of Feigenbaum Attractor*, 数学进展, 22(1993) 185.

[55] 卢钦和,广义费根鲍姆吸引子的复杂性分析,(1993)[待发表].

[56] M. J. Feigenbaum, *Quantitative universality for a class of nonlinear transformations*, J. Stat. Phys, **19**(1978) 25.

[57] E. B. Vul, Y. G. Sinai, K. M. Khanin, *Feigenbaum universality and thermodynamic formalism*, Usp. Mat. Nauk. **39**(1982) 3.

[58] M. Misiurewicz, *The structure of mappings of an interval with zero entropy*, Publ. Math. IHES **53**(1981) 5.

[59] Z. Kanfmann, *Characteristic quantities of multifractals-application to the Feigenbaum attractor*, Physica **D 54**(1991) 75.

[60] K. J. Falconer, *The Geometry of Fractals Sets*, Cambridge Univ. Press, Cambridge, 1985.

[61] A. Thue, *Über unendliche Zeichenreihen, Videnskapsselskapets Skriftere I*, Mat.-naturv. Klasse, Kristiania (1906) 1, (1912) 1.

[62] M. Morse, G. A. Hedlunds, *Unending chess, symbolic dynamics and a problem in semigroups*, Duke Math. J. **11**(1944) 1.

[63] I. Procaccia, S. Thomae, C. Tresser, *First return maps as a unified renormalization scheme for dynamical systems*, Phys. Rev. **A 35**

(1987) 1884.

[64] L. Jonker, D. Rand, *The periodic orbits and entropy of certain maps of the unit interval*, J. London Math. Soc. **22**(1980) 175.

[65] M. Garcia, G. A. Hedlund, *The structure of mimimal sets*, Bulletin of AMS **54**(1948) 954.

[66] M. Kolar, M. K. Ali, F. Nori, *Generalized Thue-Morse chains and their physical properties*, Phys, Rev. **B 43**(1991) 1034.

[67] 谢惠民,斐波那契系统的语言复杂性分析,(1993)[待发表].

[68] B. Derrida, A. Gervois, Y. Pomeau, *Iteration of endomorphisms on the real axis and representation of numbers*, Ann, Inst. Henri Poincaré **29**(1978) 305.

[69] M. Benedicks, L. Carleson, *On iteration of $1-ax^2$* on$(-1, 1)$, Ann. Math. **122**(1985) 1

[70] M. V. Jacobson, *Absolutely continuous invariant measure for one parameter families of one dimensional maps*, Commun. Math. Phys. **81**(1981)39.

[71] C. E. Shannon, *A mathematical theory of communication*, Bell System Tech. J. **27**(1948) 379.

[72] B. Mandelbrot, *On recurrent noise limiting coding*, Proc. Symposium on Information Networks, Polytechnic Institute of Brooklyn, 1955, 205.

[73] N. Chomsky, G. A. Miller, *Finite state languages*, Inform. and Control **1**(1958) 91.

[74] R. B. Banerji, *Phrase structure languages, finite machines, and channel capacity*, Inform. and Control **6**(1963) 153.

[75] W. Kuich, *On the entropy of context-free languages*, Inform. and Control **16**(1970) 173.

[76] F. P. Kaminger, *The noncomputability of the channel capacity of context-sensitive languages*, Inform. and Control **17**(1970) 175.

[77] R. Badii, *Unfolding complexity in nonlinear dynamical systems, in Measures of Complexity and Chaos*, ed. N. B. Abraham, A. M. Albano, A. Passamante, P. E. Rapp, Plenum, 1990, 313.

[78] R. Badii, M. Finardi, G. Broggi, M. A. Sepúlveda, *Hierarchical resolution of power spectra*, Physica **D 58**(1992) 304.

[79] 陈瑞熊. 陈式刚. 一维单峰映象的拓扑熵, 物理学报, 35(1986) 1338.

[80] C. S. Hsu, M. C. Kim, *Construction of maps with generating partitions for entropy evaluation*, Phys. Rev. **A 31**(1985) 3253.

[81] D. M. Cvetković, M. Doob, H. Sachs, *Spectra of Graphs-Theory and Application*, VEB, Berlin, 1982.

[82] P. Collet, J. P. Crutchfield, J.-P. Eckmann, *Computing the topological entropy of maps*, Commun. Math. Phys. **88**(1983) 257.

[83] M. Misiurewicz, W. Szlenk, *Entropy of piecewise monotone mappings*, Asterisque **50**(1977) 299.

[84] M. Misiurewicz, W. Szlenk, *Entropy of piecewise monotone mappings*, Studia Math. **67**(1980) 45.

[85] R. L. Adler, A. G. Konheim, M. H. McAndrew, *Topological entropy*, Trans. AMS **114**(1965) 309.

[86] R. Mañé, *Ergodic Theory and Differentiable Dynamics*, Springer-Verlag, New York, 1987.

[87] P. Walters, *An introduction to ergodic theory*, Springer-Verlag, New York, 1981.

[88] G. Keller, *Lyapunov exponents and complexity for interval maps*, in Lecture Notes in Math. **1486**, 1991, 216.

[89] L. Jonker, D. Rand, *The periodic orbits and entropy of certain maps of the unit interval*, J. London Math. Soc. **22**(1980) 175.

[90] 谢惠民, 陈曦, 关于某些不光滑映射的混沌带分片问题, 数学物理学报, 12(1993 增刊)65.

[91] S. N. Rasband, *Chaotic dynamics of nonlinear systems*, John Wiley and Sons, New York, 1990.

[92] S. Wolfram, *Theory and Applications of Cellular Automata*, World Scientific, Singapore, 1986.

[93] H. Gutowitz, ed., *Cellular Automata: Theory and Experiment*, Physica **D 45**, 1990.

[94] E. A. Jackson, *Perspective of nonlinear dynamnics*, V. 2, Cambridge Univ. Press, Cambridge, 1990.

[95] J. von Neumann, *Theory of Self-Reproducing Automata*, edited and completed by A. W. Burks, Univ. of Illinois Press, Champaign, Il., 1966.

[96] K. Culik II, L. P. Hurd, S. Yu, *Computation theoretic aspects of cellular automata*, Physica **D 45**(1990) 357.

[97] K. Culik II, L. P. Hurd, S. Yu, *Formal languages and global cellular automaton behavior*, Physica **D 45**(1990) 396.

[98] L. P. Hurd, J. Kari, K. Culik II, *The topological entropy of cellular automata is uncomputable*, Ergodic Theory and Dynamic Systems **12**(1992) 255.

[99] J. Kari, *The nilpotency problem of one-dimensional cellular automata*, SIAM J. Comp. **21**(1992) 571.

[100] B. Weiss, *Subshifts of finite type and sofic systems*, Monat, Math. **77**(1973) 462.

[101] O. Martin, A. M. Odlyzko, S. Wolfram, *Algebraic Properties of Cellular Automata*, Commun. Math.Phys. **93**(1984) 219.

[102] E. Jen, *Aperiodicity in one-dimensional cellular automata*, Physica **D 45**(1990) 3.

[103] L. P. Hurd, *Recursive cellular automata invariant sets*, Complex Systems **4**(1990) 119.

[104] L. P. Hurd, *Nonrecursive cellular automata invariant sets*.Complex Systems **4**(1990) 131.

[105] M. Li, P. M. B. Vitányi, *Kolmogorov complexity and its applications, in Handbook of Theoretical Computer Science*, ed. J. van Leeuwen, Elsevier Science Publishers B. V., 1990, 187.

[106] R. J. Solomonoff, *A formal theory of inductive inference*, Inform. and Control **7**(1964) 1, 224.

[107] A. N. Kolmogorov, *Three approaches to the definition of the concept 'quantity of information'*, Problem of Information Transmission, **1**(1965) 1.

[108] G. J. Chaitin, *On the length of programs for computing finite binary sequences*, J. ACM **13**(1966) 547.

[109] P. Martin-Löf, *The definition of random sequences*, Inform. and Control **9**(1966) 602.

[110] G. J. Chaitin, *Information-theoretic computational complexity*, IEEE Trans. IT-**20**(1974) 10.

[111] G. J. Chaitin, *Randomness and mathematical proof*, Scientific

American **232**(May 1975) 47.

[112] A. A. Brudno, *Entropy and the complexity of the trajectories of a dynamical system*, Trans. Moscow Math. Soc. **2**(1983) 127.

[113] 万哲先,代数和编码[修订版],科学出版社,1980.

[114] J. L. Massey, *Shift-register synthesis and BCH decoding*, IEEE Trans. IT-**15**(1969) 122.

[115] A. H. Chan, R. A. Games, E. L. Key, *On the Complexity of de Bruijn Sequences*, J. Combinatorial Theory, Series A **33** (1982) 233.

[116] R. A. Games, A. H. Chan, *A fast algorithm for determining the complexity of a binary sequence with period 2^n*, IEEE Trans. IT-**29**(1983) 144.

[117] C. J. A. Jansen, D. E. Boekee, *The shortest feedback shift register that can generate a given sequence, in Advances in Cryptology-CRYPTO'89*, 1990, 90.

[118] N. G. deBruijn, *A combinatorial problem*, Ned. Akad. Weten. Proc. **49**(1946) 758.

[119] T. Beth, Z. D. Dai, *On the complexity of pseudo-random sequences, Advances in Cryptology-EUROCRYPT'89* Springer-Verlag, 1990, 533.

[120] 鲍丰,关于线性复杂度与 Kolmogorov 复杂度之间关系的一个注记,第三届中国密码学学术会议论文集, 1992, 242.

[121] A. Lempel, J. Ziv, *On the complexity of finite sequences*, IEEE Trans. IT-**22**(1976) 75.

[122] F. Kaspar, H. G. Schuster, *Easily calculable measure for the complexity of spatiotemporal patterns*, Physical Rev. **A 36**(1987) 842.